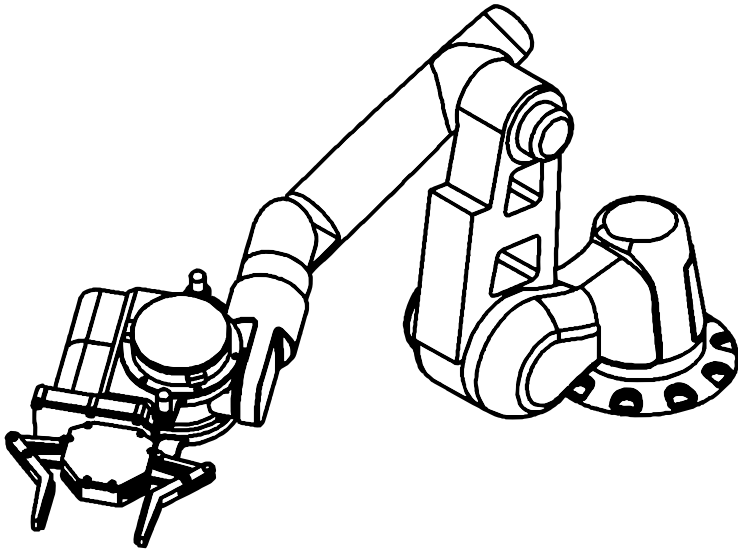


# ИТМО

Захаров Д.Н., Куровский Д.М., Ракшин Е.А.,  
Борисов О.И., Громов В.С., Колюбин С.А.

## МОДЕЛИРОВАНИЕ И УПРАВЛЕНИЕ ДВИЖЕНИЕМ РОБОТОВ



Санкт-Петербург  
2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
УНИВЕРСИТЕТ ИТМО

Захаров Д.Н., Куровский Д.М., Ракшин Е.А.,  
Борисов О.И., Громов В.С., Колубин С.А.

**МОДЕЛИРОВАНИЕ И УПРАВЛЕНИЕ  
ДВИЖЕНИЕМ РОБОТОВ**

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ  
В УНИВЕРСИТЕТЕ ИТМО

по направлениям подготовки 15.04.06 "Мехатроника и робототехника",  
27.04.03 "Системный анализ и управление"  
и 27.04.04 "Управление в технических системах"  
в качестве учебно-методического пособия для реализации основных  
профессиональных образовательных программ высшего образования  
магистратуры

**ИТМО**

Санкт-Петербург  
2023

Захаров Д.Н., Куровский Д.М., Ракшин Е.А., Борисов О.И., Громов В.С., Колобун С.А. Моделирование и управление движением роботов. Учебно-методическое пособие. — СПб.: Университет ИТМО, 2023. — 84 с.

### Рецензент:

- Шаветов С.В., к.т.н., доцент факультета систем управления и робототехники Университета ИТМО.

Учебно-методическое пособие посвящено методам моделирования многозвенных робототехнических систем и управлению их движением при помощи программного пакета Corke Robotics Toolbox для языка программирования Python. Рассмотрены решения кинематических и динамических задач по положению и скоростям. Описаны различные подходы к решению задач планирования маршрута и траектории. Представлен ряд подходов к синтезу систем управления движением многозвенных роботов.

Учебно-методическое пособие предназначено для студентов, обучающихся по направлениям подготовки 15.04.06 "Мехатроника и робототехника", 27.04.03 "Системный анализ и управление", 27.04.04 "Управление в технических системах".

# ИТМО

**Университет ИТМО** – ведущий ВУЗ России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2023

© Захаров Д.Н., Куровский Д.М., Ракшин Е.А., Борисов О.И.,  
Громов В.С., Колобун С.А., 2023

# Оглавление

Введение	4
<b>1 Теоретическая часть</b>	<b>8</b>
1.1 Кинематика	8
1.1.1 Прямая задача кинематики	11
1.1.2 Обратная задача кинематики	17
1.1.3 Кинематика скоростей	19
1.2 Динамика	24
1.3 Планирование движения	27
1.3.1 Планирование маршрута	28
1.3.2 Планирование траектории	37
1.4 Управление движением	45
1.4.1 Одномерное управление	45
1.4.2 Многомерное управление	47
1.4.3 Импедансное управление	51
<b>2 Лабораторные работы</b>	<b>54</b>
2.1 Динамическая модель многозвенного манипулятора	54
2.2 Планирование траектории многозвенного манипулятора	57
2.3 Управление манипуляционным роботом	59
2.4 Комбинированные регуляторы для траекторного управления и непрямого управления по силе	61
<b>Заключение</b>	<b>62</b>
<b>Приложение А</b>	<b>64</b>
<b>Приложение Б</b>	<b>77</b>
<b>Приложение В</b>	<b>80</b>
<b>Литература</b>	<b>85</b>

# Введение

Разработка систем управления многозвенными роботами-манипуляторами является одним из важных аспектов современной теории автоматического управления. Это объясняется высоким спросом на подобные системы со стороны промышленности.

Внедрение роботизированных производственных линий значительно влияет на снижение процента брака готовых изделий, повышает производительность и исключает человеческий фактор на этапах производственного процесса. Однако, абсолютное замещение человека робототехнической системой не имеет смысла ввиду кривой насыщения "производительность-процент автоматизации". Автоматическая производственная линия (степень автоматизации выше 80%) требует постоянного обслуживания и присутствия специалистов по ремонту. При автоматизированной системе человек участвует не как обслуживающий персонал, а как полноценная рабочая единица, непрерывно взаимодействующая с роботом. По этой причине требуется внедрение систем управления по коллаборативному взаимодействию робототехнической системы с человеком и работы с неструктурированным окружением [1].

Параллельно с описанной выше проблемой возникает задача по снижению энергопотребления робототехнической системы, что позволит повысить её автономность (мобильные роботы) и снизить затраты на электроэнергию на производственной линии (стационарные роботы). Данная задача решается как на программном уровне, так и на этапе проектирования кинематики робототехнической системы. Данное учебно-методическое пособие нацелено на помощь в освоении подходов к моделированию робототехнических систем и управлению их движением при помощи программного пакета [Corke](#)

**Robotics Toolbox**<sup>1</sup> для языка программирования Python [2].

Целью лабораторных работ, представленных в данном пособии, является знакомство и освоение подходов к моделированию роботов и управлению их движением. В процессе выполнения работ предусмотрено изучение основных понятий, определений и подходов к выводу математических моделей роботов, включая разделы кинематики и динамики. В отдельные работы вынесено рассмотрение подходов к синтезу управления движением роботов.

В рамках лабораторных работ, представленных в учебно-методическом пособии, основное внимание уделяется структуре и управлению роботами-манипуляторами, поскольку данный тип робота на сегодняшний день наиболее часто встречается в промышленных задачах [3]. Несмотря на это, рассматриваемые подходы допускается использовать для робототехнических систем другого назначения (мобильные роботы, носимые устройства).

## История развития Corke Robotics Toolbox

Первоначально Corke Robotics Toolbox (CRT) создавался для работы в пакете прикладных программ для решения задач технических вычислений MATLAB в 1991 году в рамках диссертационных исследований Питера Корка. С момента первого опубликования в 1995-1996 годах CRT развивался более 25 лет. Перенос CRT в Python был начат в 2008 году.

Актуальная версия включает в себя:

- расширенный набор функциональных возможностей MATLAB Toolbox;
- набор инструментов Spatial Math Toolbox для Python, который предоставляет описание положения и ориентации тела в пространстве как элементы специальных евклидовых SE(3) и ортогональных групп SO(3) соответственно;
- поддержку моделей, выраженных с использованием нотации Денавита-Хартенберга (стандартной и модифицированной), последовательностей элементарных преобразований и

---

<sup>1</sup><https://github.com/petercorke/robotics-toolbox-python>

деревьев твердого тела в виде расширяемого файла Unified Robotics Description Format (URDF) на языке разметки XML; возможна поддержка кинематики разветвленного, но не замкнутого или параллельного типа;

- моделирование контактного взаимодействия.

Также одним из преимуществ актуальной версии CRT является возможность работать с Robot Operating System (ROS), но при этом не зависеть от данного программного обеспечения. Это позволяет обойти ограничения ROS на версии операционной системы и Python.

CRT для Python содержит более 30-ти моделей роботов, из которых большинство является кинематическими, но некоторые имеют инерционные и фрикционные параметры. Кинематические модели могут быть описаны различными способами: в стандартной или модифицированной нотации Денавита-Хартенберга (DH, MDH), в виде строки ETS, в виде дерева твердого тела или из файла URDF.

## Преимущества CRT в Python

Python — язык программирования, известный своей универсальностью. Он поддерживает несколько парадигм программирования, включая объектно-ориентированное, императивное и функциональное программирование. Отличается простотой синтаксиса и поставляется с множеством стандартных библиотек для решения различных программных и вычислительных задач.

Весомым преимуществом Python над проприетарной средой MATLAB является его доступность. Будучи бесплатным, Python не имеет ограничений на работу кода в разных средах разработки.

## Необходимое программное обеспечение

Для выполнения лабораторных работ из данного методического пособия необходимо установить:

- Python 3 (версия 3.6 и выше);
- один из IDE или текстовый редактор на выбор (предлагаемые IDE доступны для Windows, Linux и MAC):
  - PyCharm,
  - Spyder,
  - Thonny (наиболее оптимальный вариант, т.к. автоматически устанавливает Python),
  - Visual Studio Code,
  - Jupyter Notebook;
- Corke Robotics Toolbox для Python. Для установки в консоли вводится команда `pip3 install roboticstoolbox-python`.

## Структура методического пособия

Данное учебно-методическое пособие разработано на основе авторских материалов, которые были использованы при преподавании в 2022/2023 учебном году дисциплины "Моделирование и управление движением роботов" в Университете ИТМО, и включает в себя следующие разделы:

- кинематический анализ робототехнических систем;
- динамический анализ робототехнических систем;
- планирование движения роботов;
- организация систем управления различного типа;
- лабораторные работы;
- задание и рекомендации для выполнения курсового проекта.



# Глава 1

## Теоретическая часть

### 1.1 Кинематика

Любой манипуляционный робот состоит из звеньев, соединенных в сочленение и образующих собой кинематическую пару. Среди базовых (элементарных) сочленений выделяют два типа: вращательное (поворотное) и поступательное (призматическое). Современные манипуляционные роботы могут состоять из сложных сочленений, обладающих большим количеством степеней свободы. В таком случае можно рассматривать сочленение как комбинацию вращательных и поступательных сочленений. Графическое представление типов элементарных сочленений и рабочего органа манипулятора [4] представлено на рисунке 1.1.

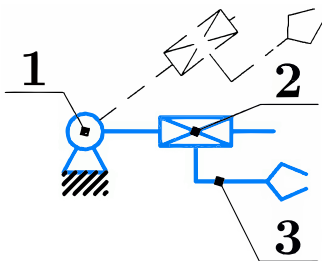


Рисунок 1.1. Типы элементарных сочленений: 1 — вращательная кинематическая пара, 2 — поступательная кинематическая пара, 3 — рабочий орган манипулятора

**Кинематическая схема** — графическое представление последовательности звеньев манипулятора.

Для вращательного сочленения относительное расположение смежных звеньев определяется угловой переменной  $\theta_i$ , а для поступательного — линейным смещением  $d_i$ . В обоих случаях эти переменные называются обобщенными координатами.

**Конфигурация манипулятора** — набор всех обобщенных координат робота, позволяющих однозначно определить его положение в пространстве.

## Анализ степени подвижности механизма

Топологию робототехнической системы с жёсткими звеньями можно описать как кинематическую конструкцию, состоящую из звеньев и кинематических пар. Кинематика механизмов и робототехнических систем разделяется на замкнутую и открытую (рис. 1.2). Открытая кинематика позволяет задать положение и ориентацию схвата через независимые степени подвижности в каждом приведённом сочленении. Для описания кинематики замкнутой системы требуется решить уравнения голономных ограничений, что усложняет кинематический и динамический анализ по сравнению с открытой кинематикой. Преимущества замкнутой кинематики над открытой: обеспечение желаемых кинематических и динамических характеристик с меньшим количеством приводов (неполноприводность); распределение внешней нагрузки на все звенья механизма; возможность вынести привод в корпусную часть робототехнической системы, снизив инерциальные нагрузки [5].

Для определения степени подвижности  $W$  плоскорычажного механизма используется формула Чебышева [6]

$$W = 3N - 2p_1 - p_2,$$

где  $N$  — число звеньев,  $p_1, p_2$  — количество низших и высших кинематических пар соответственно. Высшей называется кинематическая пара, у которой соприкосновение элементов происходит по линии или точке (кулачковые механизмы). У низкой кинематической пары контакт происходит по поверхности (ползун, плоский шарнир, винт). Количество звеньев и кинематических пар можно определить с помощью графоаналитического подхода. На рисунке 1.3 изображён пример графа одноподвижного механизма жёсткой стыковки. Количество низших кинематических пар (ветви графа)  $p_1 = 7$ , высших  $2p_2 = 0$  (на графах отображается двойной линией), количество звеньев (вершины графа)  $N = 5$ . Таким образом,  $W = 3 \cdot 5 - 2 \cdot 7 - 0 = 15 - 14 = 1$ . Единственная степень подвижности отображена на поступательной паре между 1 и 2 звеньями.

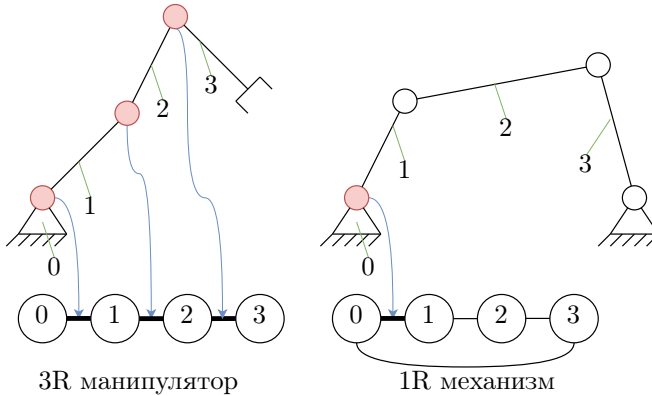


Рисунок 1.2. Механизмы открытой, замкнутой кинематики и их графоаналитическое представление

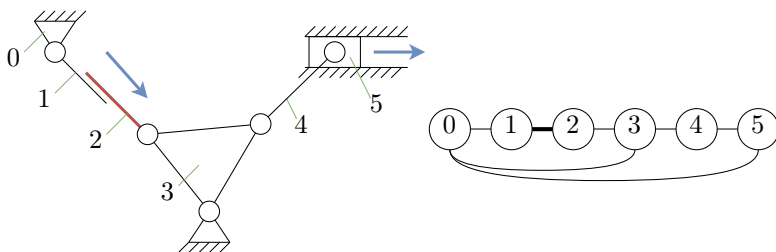


Рисунок 1.3. Определение степени подвижности механизма

При кинематическом анализе манипуляционного робота необходимо решать две основные задачи: прямую и обратную задачу кинематики [7].

### 1.1.1 Прямая задача кинематики

**Прямая задача кинематики (ПЗК)** — расчет положения и ориентации рабочего органа манипулятора (инструмента, схвата) согласно его конфигурации.

В случае двухзвенного робота для определения положения всех элементов манипулятора достаточно знать длину звеньев и угол поворота каждого звена. При переходе к реальным манипуляторам задача усложняется. Для однозначного описания расположения твердого тела в пространстве необходимо определить шесть координат: три линейные и три угловые. В элементарных случаях можно воспользоваться таким подходом, однако для реальных систем это трудоемкая задача. Одним из методов, позволяющих сократить шесть параметров, описывающих положение и ориентацию звена до четырех является использование представления Денавита-Хартенберга. Это возможно за счет стандартной процедуры присвоения систем координат звеньям манипулятора.

Алгоритм представления Денавита-Хартенберга:

Шаг 1. Присвоение систем координат к звеньям робота.

Шаг 2. Определение параметров Денавита-Хартенберга.

Шаг 3. Построение матриц однородного преобразования.

Шаг 4. Расчет углов Эйлера по итоговой матрице вращения.

Системы координат привязывают к звеньям согласно следующим правилам:

1. Ось  $z_{i-1}$  направлена вдоль оси  $i$ -го сочленения;
2. Ось  $x_i$  перпендикулярна оси  $z_{i-1}$  и пересекает ее;
3. Ось  $y_i$  выбирают так, чтобы дополнить выбранные оси до правой тройки векторов.

Параметры Денавита-Хартенберга:

- $a_i$  — расстояние вдоль оси  $x_i$  от  $z_{i-1}$  до  $z_i$ ;
- $\alpha_i$  — угол вокруг оси  $x_i$  от  $z_{i-1}$  до  $z_i$ ;
- $d_i$  — расстояние вдоль оси  $z_{i-1}$  от  $x_{i-1}$  до  $x_i$ ;
- $\theta_i$  — угол вокруг оси  $z_{i-1}$  от  $x_{i-1}$  до  $x_i$ .

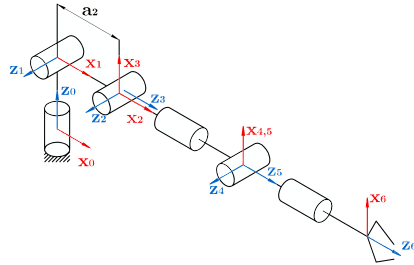
Выбор системы координат и определение параметров Денавита-Хартенберга изображены на рисунке 1.4.

Параметры Денавита-Хартенберга представлены в таблице 1.1.

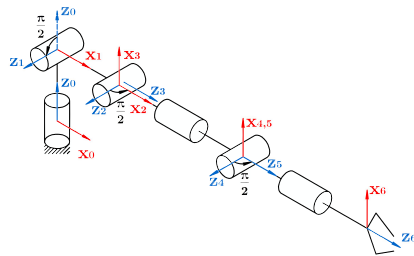
Звено, $i$	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	0	$\frac{\pi}{2}$	$d_1$	$\theta_1$
2	$a_2$	0	0	$\theta_2$
3	0	$\frac{\pi}{2}$	0	$\theta_3 + \frac{\pi}{2}$
4	0	$-\frac{\pi}{2}$	$d_4$	$\theta_4$
5	0	$\frac{\pi}{2}$	0	$\theta_5$
6	0	0	$d_6$	$\theta_6$

Таблица 1.1. Параметры Денавита-Хартенберга

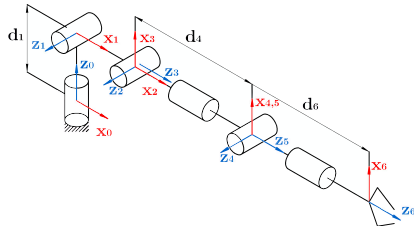
Решение ПЗК сводится к определению связи между исходной (базовой, инерциальной) системой координат  $0_0x_0y_0z_0$ , связанной



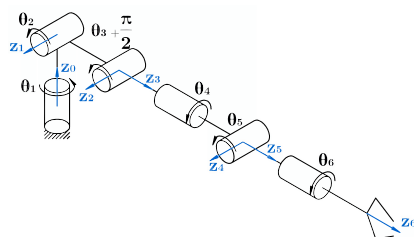
(a) Выбор осей  $z_i$



(b) Выбор осей  $x_i$



(c) Выбор осей  $y_i$



(d) Выбор системы координат  $n$

Рисунок 1.4. Алгоритм присвоения систем координат для шестизвенового манипулятора

с «землей» и итоговой  $0_n x_n y_n z_n$ , связанной со схватом или рабочим органом манипулятора. Расположение этих систем координат относительно друг друга можно определить с помощью трех угловых и трех линейных координат. Для того, чтобы перейти от одной системы координат к другой необходимо использовать матрицу однородных преобразований  $T_n^0$ .

**Матрица однородных преобразований  $T_n^0$**  — матрица, содержащая информацию о линейном смещении и пространственной ориентации одной системы координат относительно другой.

Таким образом, для определения одного набора координат  $k^0$  с помощью набора координат  $k^n$  и матрицы однородных преобразований

$$k^0 = T_n^0 k^n,$$

$$T_n^0 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_n^0 & s_n^0 & a_n^0 & p_n^0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_n^0 & p_n^0 \\ 0 & 1 \end{bmatrix},$$

где векторы  $n_n^0, s_n^0$  и  $a_n^0$  выражают направления осей  $x_n, y_n$  и  $z_n$ , относительно системы координат  $0_0 x_0 y_0 z_0$ ,  $R_n^0 \in SO(3)$  — матрица вращения системы  $0_n x_n y_n z_n$  относительно  $0_0 x_0 y_0 z_0$ ,  $p_n^0 \in \mathbb{R}^3$  — вектор линейного смещения начала координат системы  $0_n x_n y_n z_n$  относительно  $0_0 x_0 y_0 z_0$

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = T_n^0 \begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix}.$$

Подробный вывод матриц вращения можно найти в [8], самые важные свойства представлены в [9].

Базовые матрицы вращения вокруг осей  $x$ ,  $y$  и  $z$

$$R_{x,\beta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta \\ 0 & \sin \beta & \cos \beta \end{bmatrix}, \quad (1.1)$$

$$R_{y,\beta} = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}, \quad (1.2)$$

$$R_{z,\beta} = \begin{bmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.3)$$

Чтобы построить матрицы однородного преобразования, соответствующие параметрам Денавита-Хартенберга, необходимо использовать следующие преобразования

$$T_i = T_{z,\theta_i} T_{z,d_i} T_{x,a_i} T_{x,\alpha_i} = \begin{bmatrix} R_{z,\theta_i} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & p_{d_i} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & p_{a_i} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{x,\alpha_i} & 0 \\ 0 & 1 \end{bmatrix},$$

где  $i$  — номер звена,  $R_{z,\theta_i}$  и  $R_{x,\alpha_i}$  — базовые матрицы вращения, заданные (1.3) и (1.1),  $p_{d_i}$  и  $p_{a_i}$  — векторы с ненулевыми компонентам  $p_z = d_i$  и  $p_x = a_i$ , соответственно

$$p_{d_i} = \begin{bmatrix} 0 \\ 0 \\ d_i \end{bmatrix}, \quad p_{a_i} = \begin{bmatrix} a_i \\ 0 \\ 0 \end{bmatrix}. \quad (1.4)$$

Итоговую матрицу, связывающую все системы координат, можно получить последовательным перемножением

$$T_n^0(q) = T_1(q)T_2(q)\dots T_n(q) = \begin{bmatrix} R_n^0(q) & p_n^0(q) \\ 0 & 1 \end{bmatrix},$$

где матрица вращения  $R_n^0(q)$  и вектор  $p_n^0(q)$  задают, соответственно, ориентацию и положение системы координат, связанной со схватом или рабочим органом, относительно базовой системы в зависимости от конфигурации манипулятора, заданной вектором обобщенных



координат  $q \in \mathbb{R}^n$ . Вектор  $p_n^0$  имеет следующие компоненты

$$p_n^0(q) = \begin{bmatrix} x_n^0(q) \\ y_n^0(q) \\ z_n^0(q) \end{bmatrix},$$

которые являются решением ПЗК по положению в явном виде.

Матрица вращения  $R_n^0$  содержит следующие элементы

$$R_n^0(q) = \begin{bmatrix} r_{11}(q) & r_{12}(q) & r_{13}(q) \\ r_{21}(q) & r_{22}(q) & r_{23}(q) \\ r_{31}(q) & r_{32}(q) & r_{33}(q) \end{bmatrix}.$$

Для параметризации матриц вращения, состоящих из девяти элементов, могут использоваться углы Эйлера.

Матрица вращения, заданная с помощью углов Эйлера

$$R_n^0(q) = \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}.$$

Задача сводится к определению углов Эйлера  $\phi(q)$ ,  $\theta(q)$  и  $\psi(q)$  по исходной матрице вращения  $R_n^0$ . Для этого необходимо рассмотреть несколько случаев в зависимости от значения элемента  $r_{33}(q)$ :

1. Если  $r_{33}(q) \neq \pm 1$ , то

$$\begin{cases} \phi(q) = \text{atan2}(\pm r_{23}(q), \pm r_{13}(q)), \\ \psi(q) = \text{atan2}(\pm r_{32}(q), \mp r_{31}(q)). \end{cases} \quad (1.5)$$

2. Если  $r_{33}(q) = 1$ , то

$$\phi(q) + \psi(q) = \text{atan2}(r_{21}(q), r_{11}(q)).$$

3. Если  $r_{33}(q) = -1$ , то

$$\phi(q) - \psi(q) = \text{atan2}(-r_{12}(q), -r_{11}(q)). \quad (1.6)$$

Важно отметить, что при  $r_{33}(q) = \pm 1$  возникает неоднознач-

ность определения углов Эйлера. Для определенности в таких случаях принимают, например,  $\psi = 0$  и вычисляют  $\phi$  согласно (1.5) и (1.6).

Решение ПЗК для всех возможных значений обобщенных координат манипулятора ( $q$ ) позволяет определить рабочую зону манипулятора.

**Рабочая зона манипулятора** — это пространство, в котором может находиться схват манипулятора при всех возможных положениях звеньев манипулятора.

### 1.1.2 Обратная задача кинематики

**Обратная задача кинематики (ОЗК)** — расчет конфигурации манипулятора при заданном положении и ориентации рабочего органа манипулятора.

ОЗК в отличие от ПЗК, имеющей всегда единственное решение, является более сложной задачей по следующим причинам:

- неопределенность решения (одному и тому же набору положения схвата в пространстве могут соответствовать разные конфигурации робота);
- решение ОЗК не имеет единой процедуры решения и существенно зависит от конструкции манипуляционного робота.

Для решения ОЗК могут быть использованы различные методы, однако в большинстве случаев они могут быть разделены на аналитические (геометрические) и численные.

Геометрические (аналитические) методы предполагают вывод аналитических выражений, отражающих прямую зависимость координат положения и ориентации рабочего органа манипулятора и обобщенных координат с учетом конструктивных параметров робота. При этом такой метод доступен не для всех кинематических схем манипуляторов. В задачах, для которых невозможно получить аналитическое решение, применяют численные методы.

**Геометрический (аналитический) метод решения ОЗК** заключается в выводе аналитических выражений в явном виде с использованием аппарата тригонометрических функций на основе кинематической схемы манипулятора.

Заметим, что конструкция 6-звенного робота позволяет применять метод кинематической декомпозиции при решении ОЗК геометрическим методом, в связи с чем роботы такой кинематической схемы широко распространены.

Кинематическая декомпозиция заключается в разделении ОЗК на две подзадачи:

- ОЗК по положению (определение  $q_1$ ,  $q_2$  и  $q_3$ );
- ОЗК по ориентации (определение  $q_4$ ,  $q_5$  и  $q_6$ ).

**Сферическое запястье** — конструктивное расположение последних трех вращательных сочленений робота, при котором их оси вращения пересекаются в одной точке.

Алгоритм решения ОЗК для шестизвенного манипулятора со сферическим запястьем:

- 1) решить ПЗК;
- 2) на основе заданных координат рабочего органа рассчитать координаты точки пересечения осей сферического запястья;
- 3) решить ОЗК по положению, вычислив обобщенные координаты  $q_1$ ,  $q_2$  и  $q_3$  исходя из положения точки, полученной на предыдущем шаге;
- 4) получить матрицу  $R_3^0$  из решения ПЗК;
- 5) рассчитать матрицу  $R_6^3$ ;
- 6) решить ОЗК по ориентации, вычислив обобщенные координаты  $q_4$ ,  $q_5$  и  $q_6$ , как углы Эйлера, формирующие матрицу  $R_6^3$ .

Подробный пример решения ОЗК методом кинематической декомпозиции представлен в [10].

### 1.1.3 Кинематика скоростей

Кинематический анализ манипуляционного робота с помощью решения ПЗК и ОЗК позволяет установить связь между положением и ориентацией в пространстве между базовой системой координат и системой координат, связано с рабочим органом манипулятора. Однако в процессе функционирования манипулятора появляется еще одна важная характеристика — это скорость каждого звена, включая рабочий орган манипулятора. Для управления роботом необходимо установить связь между скоростью звеньев и инструмента. Важно отметить, что при наличии вращательных сочленений в кинематической схеме скорость системы координат, связанная со схватом будет иметь не только линейную, но и угловую составляющие. Отношение скорости обобщенных координат и рабочего органа можно описать, используя матрицу Якоби.

**Матрица Якоби** — матрица, связывающая линейные и угловые скорости рабочего органа манипулятора и его обобщенные скорости.

Запишем соотношение следующих векторных функций, параметризованных по времени

$$\chi(t) = J\dot{q}(t), \quad (1.7)$$

где  $\dot{q}(t) \in \mathbb{R}^n$  — вектор обобщенных скоростей,  $\chi(t) \in \mathbb{R}^6$  — агрегированный вектор линейных и угловых скоростей рабочего органа, имеющего вид

$$\chi = \begin{bmatrix} v_n^0(t) \\ \omega_{0,n}^0(t) \end{bmatrix},$$

где  $v_n^0(t) \in \mathbb{R}^3$  — вектор линейной скорости поступательного движения системы координат  $n$  относительно системы  $0$ ,  $\omega_{0,n}^0(t) \in \mathbb{R}^3$  — вектор угловой скорости вращательного движения системы координат  $n$  вокруг системы  $0$ , выраженный относительно системы  $0$ ,  $J$  — матрица Якоби манипулятора, имеющая вид

$$J = \begin{bmatrix} J_v(q(t)) \\ J_\omega(q(t)) \end{bmatrix},$$

где  $J_v(q(t))$  и  $J_\omega(q(t))$  — линейная и угловая компоненты матрицы Якоби манипулятора, соответственно.

Для вращательного и поступательного звена линейная и угловая скорость будет определяться различными значениями. Рассмотрим расчет линейной и угловой компонент матрицы Якоби для вращательного и поступательного звена.

## Компоненты матрицы Якоби манипулятора

Линейная компонента матрицы определяется как

$$J_{v_i} = \frac{\partial p_n^0}{\partial q_i},$$

которая для вращательного сочленения принимает вид

$$J_{v_i} = z_{i-1}^0,$$

а для поступательного сочленения

$$J_{v_i} = z_{i-1}^0 \times (-p_{i-1}^0 + p_n^0).$$

Сумма угловых компонентов матрицы Якоби манипулятора определяется как

$$\omega_{0,n}^0(t) = \omega_{0,1}^0 + R_1^0 \omega_{1,2}^1 + R_2^0 \omega_{2,3}^2 + \dots + R_{n-1}^0 \omega_{n-1,n}^{n-1}. \quad (1.8)$$

При этом важно отметить, что если сочленение будет поступательным, то угловая скорость соответствующего звена должна быть равна 0. Для единообразия введем бинарный параметр  $\sigma_i$ .

Для поступательного сочленения ( $\sigma_i = 0$ ) угловая скорость

$$\omega_{i-1,i}^{i-1} = 0.$$

Для вращательного сочленения ( $\sigma_i = 1$ ) угловая скорость

$$\omega_{i-1,i}^{i-1}(t) = \dot{q}_i(t)z,$$

где  $z = [0 \ 0 \ 1]^T$ .

Таким образом, (1.8) в матричной форме примет вид

$$\omega_{0,n}^0(t) = \begin{bmatrix} \sigma_1 z_0^0 & \sigma_2 z_1^0 & \sigma_3 z_2^0 & \cdots & \sigma_n z_{n-1}^0 \end{bmatrix} \begin{bmatrix} \dot{q}_1(t) \\ \vdots \\ \dot{q}_n(t) \end{bmatrix} = J_\omega q(t).$$

Таким образом, матрица Якоби для рассматриваемого 6-звенного манипулятора (см. рисунок 1.4)

$$J(q) = \begin{bmatrix} z_0^0 \times (p_6^0 - p_0^0) & z_1^0 \times (p_6^0 - p_1^0) & z_2^0 \times (p_6^0 - p_2^0) \\ z_0^0 & z_1^0 & z_2^0 \\ z_3^0 \times (p_6^0 - p_3^0) & z_4^0 \times (p_6^0 - p_4^0) & z_5^0 \times (p_6^0 - p_5^0) \\ z_3^0 & z_4^0 & z_5^0 \end{bmatrix}.$$

### Аналитическая матрица Якоби

Зачастую при решении практических задач удобно использовать производную минимальной параметризации ориентации рабочего органа с помощью трех параметров, которые в нашем случае являются углами Эйлера

$$\eta_n^0 = \begin{bmatrix} \phi(q) \\ \theta(q) \\ \psi(q) \end{bmatrix}.$$

В таком случае используют аналитическую матрицу Якоби  $J_a(q)$

$$\chi_a(t) = J_a(q)\dot{q}(t),$$

где  $\chi_a(t) \in \mathbb{R}^6$  — вектор скоростей вида

$$\chi_a(t) = \begin{bmatrix} \dot{p}_n^0(q) \\ \dot{r}_n^0(q) \end{bmatrix}.$$

Отношение между  $J(q)$  и  $J_a(q)$  можно выразить как

$$J = \begin{bmatrix} I & 0 \\ 0 & H(\eta) \end{bmatrix} J_a(q),$$

где

$$H(\eta) = \begin{bmatrix} c_\psi s_\theta & -s_\psi & 0 \\ s_\psi s_\theta & c_\psi & 0 \\ c_\theta & 0 & 1 \end{bmatrix}.$$

### Прямая и обратная задача кинематики скорости

По аналогии с ОЗК и ПЗК сформулируем определение прямой и обратной задач кинематики скорости.

**Прямая задача кинематики скорости** — расчет линейных и угловых скоростей рабочего органа при заданных обобщенных скоростях.

**Обратная задача кинематики скорости** — расчет набора обобщенных скоростей звеньев манипулятора при заданной скорости и ускорении рабочего органа манипулятора.

Выражение (1.7) является решением прямой задачи кинематики скорости.

Решение обратной задачи кинематики скорости требует рассмотрения трех случаев:

- манипуляторы с количеством звеньев  $n < 6$ , для которых уравнение (1.7) может быть разрешено относительно  $\dot{q}(t)$  при условии

$$\text{rank } J(q) = \text{rank } J_\chi,$$

где  $J_\chi = [J(q) \quad \chi]$  — расширенная матрица, тогда полученная системы линейных алгебраических уравнений может быть решена с использованием метода Гаусса;

- манипуляторы с количеством звеньев  $n = 6$ , для которых при невырожденной матрице Якоби решение ОЗК скорости имеет вид

$$\dot{q}(t) = J^{-1}(q)\chi(t);$$

- манипуляторы с количеством звеньев  $n > 6$ , для которых

необходимо вместо обратной матрицы использовать псевдообратную

$$\dot{q}(t) = J^+ \chi + (I - J^+ J) \rho,$$

где  $J^+ = J^T (J J^T)^{-1}$  — правая псевдообратная матрица от  $J$ ,  $\rho \in \mathbb{R}^n$  — произвольный вектор.

Таким образом, матрица Якоби необходима для решения задач кинематики скоростей. Заметим, что в некоторых конфигурациях матрица Якоби может становиться необратимой в связи с нулевым значением определителя, что соответствует сингулярной конфигурации.

**Сингулярные конфигурации** — определенный набор обобщенных координат, при котором дальнейшее движение схвата манипулятора затруднено в связи с тем, что ограниченные значения скорости, силы и вращающего момента рабочего органа могут соответствовать неограниченным значениям обобщенных скоростей и вращающих моментов.

В случае сингулярной конфигурации матрица Якоби  $J(q)$  вырождается (теряет ранг)

$$\text{rank } J(q_s) < \max_q \{\text{rank } J(q)\}.$$

## Вопросы для самоконтроля

1. Чем отличается замкнутая и разомкнутая кинематические схемы? Приведите примеры таких структур.
2. Дайте определение прямой задаче кинематики. Сколько решений может иметь такая задача?
3. С какой целью применяется представление Денавита-Хартенберга? Назовите параметры, используемые для данного представления.
4. Дайте определение обратной задаче кинематики, сколько решений может иметь такая задача?



5. Что такое сингулярная конфигурация? В каких условиях она может возникнуть? Приведите пример сингулярной конфигурации из механики.

## 1.2 Динамика

Динамика позволяет определить взаимную связь между движением механической системы и силами, прилагаемыми к системе.

Динамические свойства системы определяются как взаимодействием с окружающей средой и внешними воздействиями (например, силы взаимодействия, гравитационные силы, внешние нагрузки), так и внутренними динамическими свойствами звеньев, связанных с физическими и конструктивными характеристиками робота [11]. Для снижения нагрузки на приводную систему робота может использоваться специальная система противовеса.

Для расчёта инерциальных характеристик звена в пространстве приводится его тензор инерции, определяющий соотношение инерций по всем координатным осям

$$\mathcal{I}_{full} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}. \quad (1.9)$$

Расчёт такой матрицы осложняется тем, что на практике звено робота не является идеальным сплошным телом простой формы. Выпуклости, технологические полости, различные материалы в сборке звена определяют функцию плотности в тензоре, что осложняет ее аналитическое описание. Для упрощения расчётов систему координат звена располагают в центра масс, в результате чего тензор инерции принимает вид

$$\mathcal{I}_c = \text{diag}(I_x, I_y, I_z), \quad (1.10)$$

где  $I_x, I_y, I_z$  — моменты инерции по соответствующим осям.

Например, для эллипсоида с длинами осей  $a, b, c$  и массой  $M$

тензор инерции в центре масс равен

$$\mathcal{I}_c = \begin{bmatrix} \frac{1}{5}M(b^2 + c^2) & 0 & 0 \\ 0 & \frac{1}{5}M(a^2 + c^2) & 0 \\ 0 & 0 & \frac{1}{5}M(a^2 + b^2) \end{bmatrix}. \quad (1.11)$$

Для описания динамики робототехнической системы с открытой кинематикой в конфигурационном пространстве рассмотрим матричное уравнение вида

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = J^\top f = \tau, \quad (1.12)$$

где  $q, \dot{q}, \ddot{q} \in R^n$  — векторы соответственно обобщённых координат, скоростей и ускорений,  $M(q), C(q, \dot{q}), G(q)$  — соответственно матрица инерции, матрица кориолисовых и центробежных сил и вектор гравитационных сил,  $J$  — матрица Якоби, связывающая обобщённые силы в конфигурационном ( $\tau$ ) и операционном ( $f$ ) пространствах.

**Прямая задача динамики** — вычисление обобщённых координат, скоростей и ускорений  $q, \dot{q}, \ddot{q}$  на основе известных приложенных к робототехнической системе сил  $\tau$ .

**Обратная задача динамики** — поиск сил  $\tau$ , необходимых для достижения задаваемых обобщённых координат, скоростей и ускорений  $q, \dot{q}, \ddot{q}$ .

Вывод элементов из уравнения динамики (1.12) может быть выполнен с помощью уравнения Лагранжа второго рода

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \tau, \quad (1.13)$$

где  $q$  — вектор обобщённых координат,  $\mathcal{L} = \mathcal{K} - \mathcal{P}$  — Лагранжиан, равный разнице полной кинетической энергии и полной потенциальной энергии,  $\tau$  — вектор обобщённых сил. В общем виде кинетическая и потенциальная энергии тела равны

$$\mathcal{K} = \frac{1}{2}m |V|_c^2 + \frac{1}{2}\omega^\top \mathcal{I}\omega, \quad \mathcal{P} = mgh_c. \quad (1.14)$$

где  $m$  — масса тела,  $V_c$ ,  $\omega$  — линейная и угловая скорости тела,  $\mathcal{I}$  — тензор инерции относительно центра масс,  $h_c$  — высота центра масс относительно инерциальной системы координат.

Рассмотрим материальную точку массы  $m$ , которая свободно падает вдоль оси  $Oy$  с высоты  $h$ . Кинетическая энергия, потенциальная энергия и Лагранжиан равны

$$\mathcal{K} = \frac{1}{2}mV^2, \quad \mathcal{P} = mgh, \quad \mathcal{L} = \frac{1}{2}mV^2 - mgh. \quad (1.15)$$

Первое и второе слагаемых левой части выражения (1.13) принимают вид

$$\begin{aligned} \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} &= \frac{d}{dt} \frac{\partial}{\partial \dot{y}} \left( \frac{1}{2}m\dot{y}^2 - mgy \right) = m\ddot{y}, \\ -\frac{\partial \mathcal{L}}{\partial q} &= -\frac{\partial}{\partial y} \left( \frac{1}{2}m\dot{y}^2 - mgy \right) = mg, \end{aligned} \quad (1.16)$$

с учетом которых имеем

$$m\ddot{y} + mg = 0. \quad (1.17)$$

При наличии диссипативных сил (трение, нагрев) правая часть выражения (1.17) не равна нулю и определяется диссипативной функцией  $R$  как

$$m\ddot{y} + mg = \frac{\partial R}{\partial \dot{y}}. \quad (1.18)$$

Уравнение (1.17) может быть представлено в форме (1.12) как

$$M(q) = m, \quad C(q, \dot{q}) = 0, \quad G(q) = mg. \quad (1.19)$$

Поскольку кориолисовы и центробежные силы связаны с взаимным движением двух тел и вращательным движением, для материальной точки, свободно падающей вдоль вертикальной оси, составляющая  $C(q, \dot{q})$  равна нулю.

Для аналитического описания модели динамики механизмов, содержащих замкнутые кинематические цепи, требуется определить голономные ограничения вида

$$N_c(q) = 0. \quad (1.20)$$

Дифференцируя выражение (1.20), найдем матрицу Якоби сил связей

$$J_c \dot{q} = 0. \quad (1.21)$$

В результате уравнение динамики для роботов и механизмов с замкнутыми кинематическими цепями принимает вид

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = S^\top \tau + J^\top f_{ext} + J_c^\top \lambda, \quad (1.22)$$

где  $S$  — бинарная матрица, определяющая в какой кинематической паре находится двигатель,  $J$  — матрица Якоби, преобразующая силы  $f_{ext}$  из операционного пространства в конфигурационное,  $J_c$  — матрица Якоби сил связей  $\lambda$ .

## Вопросы для самоконтроля

1. В чем заключается задача одномерного управления манипуляционным роботом?
2. Сформулируете задачу стабилизации для двухзвенного робота.
3. Сформулируете задачу слежения за траекторией для многозвенного робота.
4. Назовите не менее двух методов настройки ПИД регулятора и опишите этапы настройки.
5. В каких реальных промышленных задачах может применяться импедансное управление манипуляционным роботом?

## 1.3 Планирование движения

Для выполнения той или иной полезной операции роботом необходимо определить его желаемое поведение и формализовать цель управления [12]. Для решения задачи управления движением необходимо определить заданное (желаемое) движение. Как правило, эта задача решается в два этапа. Первый этап заключается в поиске последовательности опорных точек, определяющих маршрут

(путь) движения, а второй — в поиске функции времени, определяющей траекторию движения, то есть параметризованную по времени функцию, проходящую через выбранные на первом этапе опорные точки и интерполирующую значения между ними.

Задача планирования маршрута можно выполняться либо в рабочем (декартовом) или в конфигурационном пространствах [13]. Для многосвязных манипуляционных роботов целесообразно планировать движение в конфигурационном пространстве  $Q = \{q\}$ .

**Конфигурационное пространство** — множество всех возможных конфигураций робота.

### 1.3.1 Планирование маршрута

Рассмотрим некоторые известные методы, применяемые для планирования маршрута движения манипуляционного робота.

**Планированием маршрута (пути) в конфигурационном пространстве** — процесс поиска последовательности опорных конфигураций от начальной до заданной с обходом запрещенных областей, соответствующих столкновению робота с препятствиями или с собственными звеньями.

#### Методы на основе графов

Перечислим некоторые ключевые методы на основе графов.

1. Диаграмма видимости — граф взаимной видимости точек пространства. Для диаграммы видимости необходимо, чтобы препятствия имели форму многоугольника или многогранника в зависимости от размерности пространства [14].
2. Диаграмма Вороного — метод планирования путем разбиения  $n$ -мерного пространства (в частном случае для задач планирования рассматривается плоскость) на  $m$  локусов (область, в

которой присутствует некоторое количество близлежащих точек). Данный алгоритм позволяет получить более безопасный путь при увеличении общей длины [15].

3. Вероятностная дорожная карта — метод, основанный на использовании случайных выборок в конфигурационном пространстве. Применяется для быстрой генерации пути (см. рисунок 1.5) [16].
4. Метод быстро исследующих деревьев (RRT) — метод на основе случайного выбора. Генерация дерева (пути) состоит в построении опорных точек, которые расширяются от начальной до целевой точки (возможны обратные варианты, когда дерево строится из целевой точки). Данный алгоритм применим в конфигурационном пространстве [17].

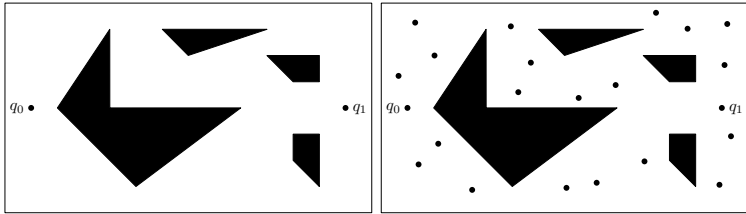
Описанные методы определяют опорные значения положений, скоростей и ускорений, которые должен принимать робот в каждом узле графа. Переходы между узлами характеризуются функцией затрат, что с учетом соответствующего критерия оптимизации позволяет выбрать наиболее оптимальный маршрут, характеризующийся минимальными затратами для достижения целевого положения.

Рассмотренную группу алгоритмов рекомендовано использовать в статической полностью известной окружающей среде. Наличие динамических препятствий (другой робот, человек и т.д.) при данном подходе существенно осложняет построение графа.

## **Метод на основе клеточной декомпозиции**

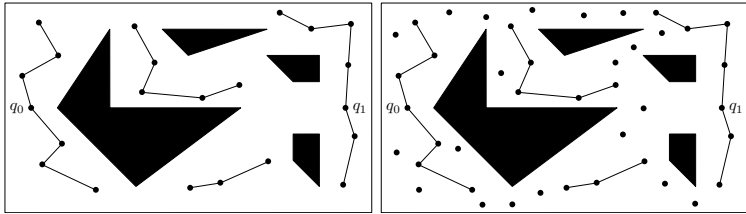
Метод декомпозиции на ячейки является наиболее простым и используется для двухмерного пространства [18], что мотивирует использовать данный алгоритм для наземных и надводных мобильных роботов. Различают точную и приближительную декомпозицию на ячейки.

Принцип точной декомпозиции заключается в разделении всего свободного конфигурационного пространства на треугольные или трапецидальные ячейки и составлении графа связности с вершинами в их центрах и ребрами, соответствующими общим сторонам



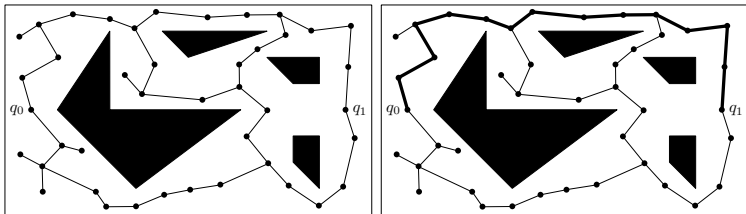
(a) Конфигурационное пространство  $Q$

(b) Случайная выборка 20 конфигураций



(c) Соединение ближайших узлов с помощью отрезков

(d) Повторная выборка 20 конфигураций



(e) Повторное соединение ближайших узлов отрезками

(f) Определение последовательности узлов, соединяющей начальную и конечную конфигурации

Рисунок 1.5. Планирование маршрута с помощью метода вероятностной дорожной карты

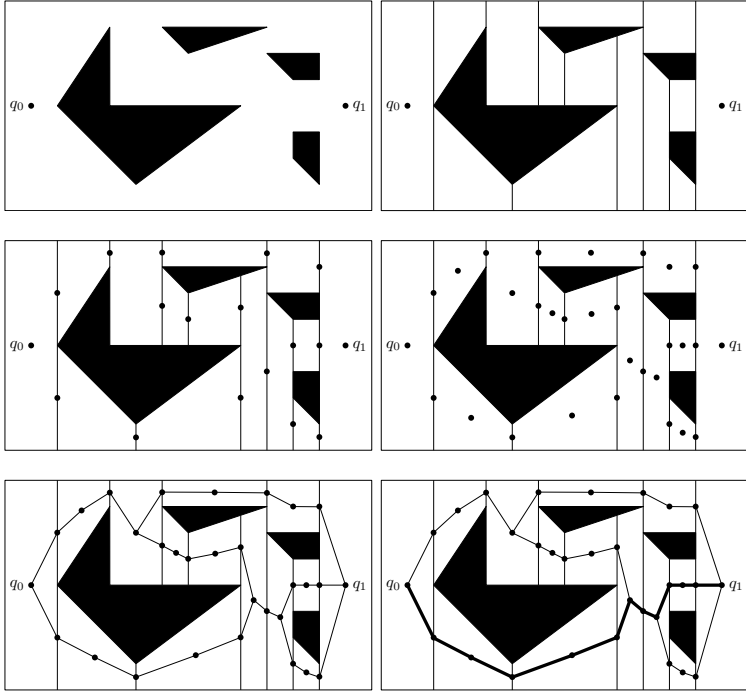


Рисунок 1.6. Этапы принципа точной декомпозиции

смежных ячеек (см. рисунок 1.6). Запрещенные области при этом представляют собой многоугольники.

В точной декомпозиции используются два типа ячеек:

- белые, соответствующие свободному конфигурационному пространству;
- черные, соответствующие запрещенной области конфигурационного пространства.

Суть метода заключается в задании вершин графа, соответствующим начальной и конечной конфигурациям, и в поиске последовательности переходов по белым ячейкам, соединяющей эти две вершины.



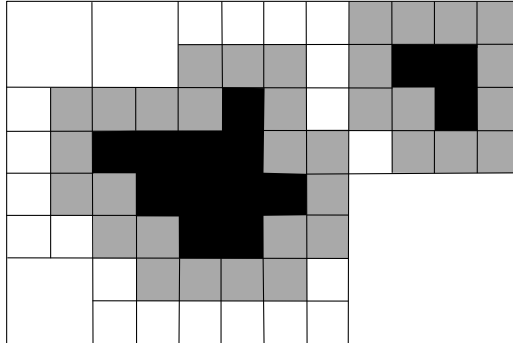


Рисунок 1.7. Приблизительная декомпозиция ячейки

Приблизительную декомпозицию отличает наличие ячеек, которые содержат как свободные, так и запрещенные области (см. рисунок 1.7), что может быть обусловлено сложной формой запрещенной зоны [19]. В приблизительной декомпозиции используются три типа ячеек:

- белые, соответствующие свободному конфигурационному пространству;
- серые, содержащие как свободные, так и запрещенные области;
- черные, соответствующие запрещенной области конфигурационного пространства.

При этом в процессе поиска планируемый маршрут может проходить через белые и серые ячейки. В случае попадания на серую осуществляется ее декомпозиция до тех пор, пока весь маршрут не окажется на белых ячейках.

Ключевой недостаток методов построенных на клеточной декомпозиции заключается в значительном увеличении сложности построения пути при уменьшении шага сетки [20].

## Методы на основе потенциальных полей

Метод потенциального поля заключается в том, что робот рассматривается как материальная точка, движущаяся в конфигурационном пространстве под воздействием некоторой функции потенциального поля  $P(q)$  [21]. Эта функция имеет attractive компоненту  $P_a(q)$  (область притяжения), соответствующую конечной конфигурации робота, и repulsive компоненту  $P_r(q)$  (области отталкивания), соответствующую границам запрещенных областей конфигурационного пространства, обусловленных наличием препятствий:

$$P(q) = P_a(q) + P_r(q).$$

В качестве attractive компоненты  $P_a(q)$  задается глобальный минимум функции  $P(q)$ :

$$P_a(q) = \frac{1}{2}k_a\|q - q_d\|^2,$$

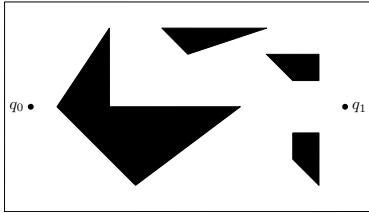
где  $q, q_d$  — текущая и желаемая конфигурации манипулятора,  $k_a$  — масштабирующий коэффициент. Repulsive компонента  $P_r(q)$  обеспечивает обращение функции в бесконечность при приближении материальной точки на определенное расстояние к препятствию:

$$P_r(q) = \begin{cases} \frac{1}{2}k_r \left( \frac{1}{\delta(q)} - \frac{1}{\delta_0} \right)^2 & \text{if } \delta(q) \leq \delta_0, \\ 0 & \text{if } \delta(q) > \delta_0, \end{cases}$$

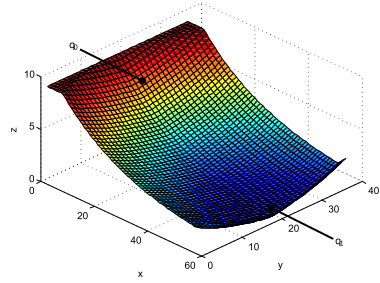
где  $k_r$  — масштабирующий коэффициент,  $\delta(q)$  — кратчайшее расстояние от текущей конфигурации  $q$  до границы запрещенной области,  $\delta_0$  — минимальное допустимое значение  $\delta(q)$ .

Планирование маршрута таким методом для манипуляционного робота предполагает итерационный расчет конфигурационного пространства в процессе поиска пути (см. рисунок 1.8) [22].

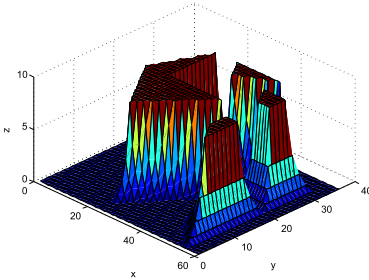
Планирование маршрута сводится к решению оптимизационной задачи поиска глобального минимума функции, начиная от исходной конфигурации  $q_0$ . В качестве способа решения такой задачи может быть использован широко известный метод градиентного спуска, обеспечивающий движение материальной точки в направлении,



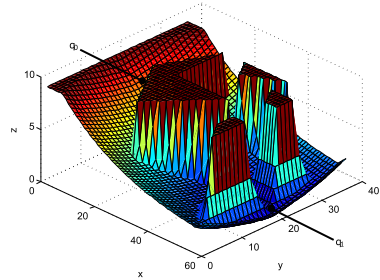
(a) Конфигурационное пространство  $Q$



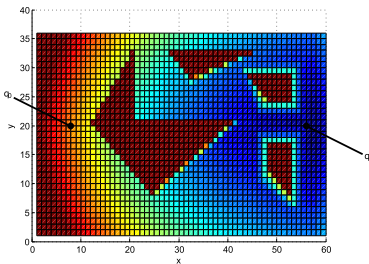
(b) Задание аттрактивной компоненты потенциального поля



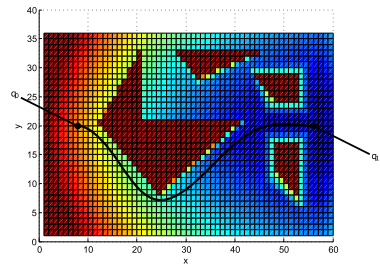
(c) Задание репульсивной компоненты потенциального поля



(d) Комбинирование аттрактивной и репульсивной компонентов



(e) Решение оптимизационной задачи поиска глобального минимума функции



(f) Определение маршрута соединяющего начальную и конечную конфигурации

Рисунок 1.8. Планирование маршрута с помощью метода потенциального поля

заданном антиградиентом функции  $-\nabla P(q)$

$$q_{j+1} = q_j - \gamma_j \nabla P(q_j),$$

где  $\nabla P(q) = \left[ \frac{\partial P}{\partial q_1} \quad \frac{\partial P}{\partial q_2} \cdots \frac{\partial P}{\partial q_n} \right]^T$ ,  $\gamma_j$  — шаг, который может быть либо постоянным, либо дробным, либо обусловленным наискорейшим спуском

$$\gamma_j = \operatorname{argmin}_j P(q_j - \gamma \nabla P(q_j)).$$

Недостатком метода градиентного спуска и, как следствие, метода потенциального поля в целом является риск попасть в локальный минимум гиперповерхности конфигурационного пространства вместо глобального минимума [23]. Для устранения этого недостатка алгоритм может быть дополнен так называемым методом случайного блуждания [24].

## Оптимизационные методы

Известным подходом к решению задачи планирования пути является рассмотрение ее как оптимизационной задачи [25]. При этом движения робота описываются в виде динамической системы, препятствия — в виде ограничений, а качество маршрута будет оцениваться некоторым функционалом. Решение такой задачи позволит не только спланировать путь в обход препятствий, но и выбрать наиболее приемлемый маршрут, например, с учетом оцениваемой скорости прохождения или энергетической эффективности исполнительных двигателей робототехнической системы [26].

В общем случае задачу оптимального управления можно представить следующим образом. Требуется найти решение  $x(t)$ ,  $u(t)$  динамической системы

$$\dot{x} = f(x, u, t)$$

при заданных ограничениях:

- $l_i \leq \psi_i(x(t_0), u(t_0), t_0) \leq u_i$  (ограничения в начальной точке);
- $l_t \leq \psi_t(x(t), u(t), t) \leq u_t$  (траекторные ограничения);
- $l_f \leq \psi_f(x(t_f), u(t_f), t_f) \leq u_f$  (конечные ограничения).

При этом решение должно обеспечивать минимизацию целевого функционала

$$J[x(t), u(t)] = \int_{t_0}^{t_f} \phi_t(x(t), u(t), t) dt + \phi_0(x(t_0), u(t_0), t_0) + \phi_f(x(t_f), u(t_f), t_f) \rightarrow \min .$$

Существуют различные подходы, направленные на решение обозначенных задач: параметризация плоских выходов, смешанное целочисленное линейное программирование, двоично-целочисленное программирование. Оптимальное планирование пути является нетривиальной задачей, особенно для нелинейных систем (к которым относятся робототехнические приложения). Для расчета оптимального маршрута нелинейных систем в основном прибегают к численным методам, которые различаются на прямые и косвенные [27]. Таким образом, основным недостатком оптимальных алгоритмов является трудоемкость их реализации, особенно применительно к реальным работам.

## Методы на основе биоинспирированного подхода

Рассмотрим методы на основе биоинспирированного подхода для планирования маршрутов движения роботов.

**Биоинспирированный подход** — подход, заимствующий (имитирующий) поведение или мышление человека или других биологических групп.

Рассмотрим некоторые алгоритмы на основе этого подхода.

1. Муравьиный метод — метод, основанный на поведении муравьев, отправляющихся из своих колоний на поиск пищи. Использование муравьиного метода можно рассматривать как близкое к оптимальному решение сложной комбинаторной задачи за допустимое время. Данный метод применим как к двумерной, так и к трехмерной постановке задачи [28, 29];

2. Метод роя частиц — метод псевдооптимального поиска пути с помощью частиц перемещающихся по определенному закону в разрешенной зоне (имитирующих поведение птиц в природе) без использования точного градиента оптимизируемой функции [30];
3. Реактивные методы — методы, основанные на имитации поведенческой реакции, т.е. решения для типовых задач [31]. При этом каждый «решатель» для такой задачи является отдельным блоком внутри системы. При таком методе чаще всего используются такие блоки, как движение к цели, обход препятствия, движение вдоль стены и т.д.

Все рассмотренные методы не исчерпывают подходы к решению задач планирования пути. Многие из них имеют множество модификаций и применяются совместно с другими методами. Ведутся исследования и разработки, направленные на улучшение скорости, точности, энергоэффективности и коллаборативности алгоритмов поиска маршрута для различных технических систем.

### 1.3.2 Планирование траектории

Задача планирования траектории заключается в расчете функций от времени, определяющих значения положений, скоростей и ускорений движения робота при перемещении по заданному маршруту движения.

**Планирование траектории** — процесс поиска функции времени, параметризующей перемещение робота по заданному маршруту движения.

Широко известным методом планирования траекторий является метод интерполяции с помощью сплайн-функций, описанный в [24, 32]. Данный подход заключается в расчете промежуточных значений обобщенных координат, скоростей и ускорений для каждого звена между заданными конфигурациями с помощью полиномов

вида

$$\begin{aligned} q_i(t) &= a_{l,i}t^l + a_{l-1,i}t^{l-1} + \dots + a_{2,i}t^2 + a_{1,i}t + a_{0,i}, \\ \dot{q}_i(t) &= la_{l,i}t^{l-1} + (l-1)a_{l-1,i}t^{l-2} + \dots + 2a_{2,i}t + a_{1,i}, \\ \ddot{q}_i(t) &= l(l-1)a_{l,i}t^{l-2} + (l-1)(l-2)a_{l-1,i}t^{l-3} + \dots + 2a_{2,i}, \end{aligned}$$

где степень  $l$  и коэффициенты  $a_{ji}$ ,  $j = \{1, 2, \dots, l\}$  определяются в зависимости от граничных условий и наложенных требований к непрерывности траектории.

Данные полиномы задаются для каждой обобщенной координаты  $i$ . Далее в рассматриваемых примерах индекс  $i$  опущен для краткости и приняты следующие обозначения

$$q_i(\cdot) \equiv \vartheta(\cdot), \quad \dot{q}_i(\cdot) \equiv v(\cdot), \quad \ddot{q}_i(\cdot) \equiv \alpha(\cdot).$$

Исходными данными для планирования траектории являются конфигурации опорных точек маршрута и моменты времени их прохождения. Рассмотрим общий алгоритм планирования траекторий с помощью сплайн-функций:

- 1) определение количества сегментов, на которые разделяется исходная траектория (в случае задания промежуточных конфигураций между исходной и конечной);
- 2) нормирование времени (в случае многосегментной траектории);
- 3) определение граничных условий и требований на непрерывность траектории между смежными сегментами;
- 4) определение степеней полиномов для каждого звена и сегмента траектории;
- 5) составление матричного уравнения и разрешение его относительно неизвестных коэффициентов полиномов.

### Односегментная траектория

Рассмотрим задачу перемещения манипуляционного робота из начального положение в конечное. При этом в качестве исходных дан-

ных выступают только граничные условия, т.е. обобщенные координаты, скорости и ускорения в начальный и конечный моменты времени. Вся траектория при этом будет представлять собой единый сегмент, поскольку никаких промежуточных требований не накладывается. Таким образом, нормирование времени производить необязательно ввиду относительной простоты вычислений.

Запишем следующие граничные условия для каждого звена  $i$

$$q_i(t_0) = \vartheta_0, \quad \dot{q}_i(t_0) = v_0, \quad \ddot{q}_i(t_0) = \alpha_0, \quad (1.23)$$

$$q_i(t_1) = \vartheta_1, \quad \dot{q}_i(t_1) = v_1, \quad \ddot{q}_i(t_1) = \alpha_1. \quad (1.24)$$

Поскольку имеется шесть условий, т.е. шесть известных значений, необходимо выбрать такой вид полинома, который будет содержать соответственно шесть неизвестных коэффициентов, чтобы итоговое матричное уравнение было разрешимо. Для рассматриваемого случая выберем полином пятой степени с шестью коэффициентами

$$\vartheta(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0.$$

Возьмем первую и вторую производные от полинома для интерполяции скорости и ускорения

$$\dot{\vartheta}(t) = v(t) = 5a_5 t^4 + 4a_4 t^3 + 3a_3 t^2 + 2a_2 t + a_1,$$

$$\ddot{\vartheta}(t) = \alpha(t) = 20a_5 t^3 + 12a_4 t^2 + 6a_3 t + 2a_2.$$

Теперь составим систему из шести уравнений, соотнося полиномы с граничными условиями (1.23)-(1.24)

$$\left\{ \begin{array}{l} \vartheta_0 = a_5 t_0^5 + a_4 t_0^4 + a_3 t_0^3 + a_2 t_0^2 + a_1 t_0 + a_0, \\ v_0 = 5a_5 t_0^4 + 4a_4 t_0^3 + 3a_3 t_0^2 + 2a_2 t_0 + a_1, \\ \alpha_0 = 20a_5 t_0^3 + 12a_4 t_0^2 + 6a_3 t_0 + 2a_2, \\ \vartheta_1 = a_5 t_1^5 + a_4 t_1^4 + a_3 t_1^3 + a_2 t_1^2 + a_1 t_1 + a_0, \\ v_1 = 5a_5 t_1^4 + 4a_4 t_1^3 + 3a_3 t_1^2 + 2a_2 t_1 + a_1, \\ \alpha_1 = 20a_5 t_1^3 + 12a_4 t_1^2 + 6a_3 t_1 + 2a_2. \end{array} \right.$$



Запишем эту систему уравнений в матричном виде

$$\begin{bmatrix} \vartheta_0 \\ v_0 \\ \alpha_0 \\ \vartheta_1 \\ v_1 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} t_0^5 & t_0^4 & t_0^3 & t_0^2 & t_0 & 1 \\ 5t_0^4 & 4t_0^3 & 3t_0^2 & 2t_0 & 1 & 0 \\ 20t_0^3 & 12t_0^2 & 6t_0 & 2 & 0 & 0 \\ t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1 & 1 \\ 5t_1^4 & 4t_1^3 & 3t_1^2 & 2t_1 & 1 & 0 \\ 20t_1^3 & 12t_1^2 & 6t_1 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix},$$

откуда выразим вектор неизвестных коэффициентов

$$\begin{bmatrix} a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} t_0^5 & t_0^4 & t_0^3 & t_0^2 & t_0 & 1 \\ 5t_0^4 & 4t_0^3 & 3t_0^2 & 2t_0 & 1 & 0 \\ 20t_0^3 & 12t_0^2 & 6t_0 & 2 & 0 & 0 \\ t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1 & 1 \\ 5t_1^4 & 4t_1^3 & 3t_1^2 & 2t_1 & 1 & 0 \\ 20t_1^3 & 12t_1^2 & 6t_1 & 2 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \vartheta_0 \\ v_0 \\ \alpha_0 \\ \vartheta_1 \\ v_1 \\ \alpha_1 \end{bmatrix}.$$

Таким образом, мы получили полиномы для каждого звена, рассчитали их коэффициенты с помощью которых может быть осуществлена интерполяция обобщенных координат, скоростей и ускорений при перемещении робота из одной точки в другую.

## Многосегментная траектория

Задача построения траектории для манипуляционного робота усложняется, когда помимо граничных условий заданы промежуточные конфигурации. Например, типичной задачей для робота является перемещение манипулируемого объекта из одной рабочей зоны в другую. Таким образом, мы можем разделить траекторию на три сегмента, границами которых будут являться заданные конфигурации манипулятора (см. рисунок 1.9):

- сегмент ухода (*a*);
- сегмент собственно перемещения между рабочими зонами (*b*);
- сегмент подхода (*c*).

В соответствии с исходными данными запишем условия на обобщенные координаты, скорости и ускорения для каждого звена *i* (в

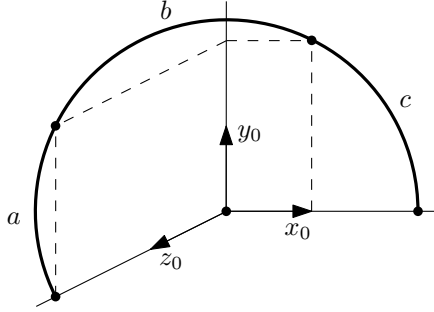


Рисунок 1.9. Траектория, разделенная на 3 сегмента

дальнейшем индекс опущен)

$$q_i(t_0) = \vartheta_0, \quad q_i(t_1) = \vartheta_1, \quad q_i(t_2) = \vartheta_2, \quad q_i(t_3) = \vartheta_3, \quad (1.25)$$

$$\dot{q}_i(t_0) = v_0, \quad \ddot{q}_i(t_0) = \alpha_0, \quad \dot{q}_i(t_3) = v_3, \quad \ddot{q}_i(t_3) = \alpha_3. \quad (1.26)$$

При нескольких сегментах внутри траектории для унификации расчетов удобно использовать нормированное время для каждого из сегментов

$$\tau_a = \frac{t - t_0}{t_1 - t_0}, \quad \tau_b = \frac{t - t_1}{t_2 - t_1}, \quad \tau_c = \frac{t - t_2}{t_3 - t_2}, \quad (1.27)$$

где  $t_0, t_1, t_2, t_3$  — моменты времени прохождения заданных конфигураций.

Для каждого из сегментов необходимо задать соответствующие полиномы  $\vartheta_a(\tau_a)$ ,  $\vartheta_b(\tau_b)$  и  $\vartheta_c(\tau_c)$  для интерполяции значений обобщенных координат. Имеющиеся сегменты необходимо «склеить» между собой, чтобы обеспечить непрерывность движения на протяжении всей траектории. Для этого, помимо условий (1.25) и (1.26), вводят требования на непрерывность. Они заключаются в том, что скорость и ускорение в конце и начале смежных сегментов траектории должны быть равны

$$v_a(1) = v_b(0), \quad \alpha_a(1) = \alpha_b(0), \quad v_b(1) = v_c(0), \quad \alpha_b(1) = \alpha_c(0). \quad (1.28)$$

Далее с учетом нормированного времени (1.27) перепишем условия на обобщенные координаты (1.25)

$$\vartheta_a(0) = \vartheta_0, \quad \vartheta_b(0) = \vartheta_1, \quad \vartheta_c(0) = \vartheta_2, \quad (1.29)$$

$$\vartheta_a(1) = \vartheta_1, \quad \vartheta_b(1) = \vartheta_2, \quad \vartheta_c(1) = \vartheta_3 \quad (1.30)$$

и граничные условия на скорости и ускорения (1.26)

$$v_a(0) = v_0, \quad \alpha_a(0) = \alpha_0, \quad v_c(1) = v_3, \quad \alpha_c(1) = \alpha_3. \quad (1.31)$$

Таким образом, получилось 14 условий (1.28)-(1.31). Теперь необходимо выбрать такие полиномы  $\vartheta_a(\tau_a)$ ,  $\vartheta_b(\tau_b)$  и  $\vartheta_c(\tau_c)$ , чтобы их совокупное число коэффициентов было равно 14 для того, чтобы матричное уравнение было разрешимо. Поскольку всего полиномов необходимо выбрать три (для каждого элементарного сегмента) и каждый из них имеет свободный член, суммарное количество старших степеней полиномов должно быть равно 11. Соответственно, допустимы такие комбинации полиномов, как 4-3-4, 3-5-3 и т.д.

Выберем полиномы для описания изменения обобщенных координат на каждом из сегментов

$$\begin{aligned} \vartheta_a(\tau_a) &= a_4\tau_a^4 + a_3\tau_a^3 + a_2\tau_a^2 + a_1\tau_a + a_0, \\ \vartheta_b(\tau_b) &= b_3\tau_b^3 + b_2\tau_b^2 + b_1\tau_b + b_0, \\ \vartheta_c(\tau_c) &= c_4\tau_c^4 + c_3\tau_c^3 + c_2\tau_c^2 + c_1\tau_c + c_0. \end{aligned}$$

Далее процесс решения совпадает с расчетом односегментной траектории. Система из 14 уравнений с учетом налагаемых условий принимает вид

$$\left\{ \begin{array}{l} \vartheta_0 = a_0, \\ v_0 = a_1, \\ \alpha_0 = 2a_2, \\ \vartheta_1 = a_4 + a_3 + a_2 + a_1 + a_0, \\ \vartheta_1 = b_0, \\ 0 = 4a_4 + 3a_3 + 2a_2 + a_1 - b_1, \\ 0 = 12a_4 + 6a_3 + 2a_2 - 2b_2, \\ \vartheta_2 = b_3 + b_2 + b_1 + b_0, \\ \vartheta_2 = c_0, \\ 0 = 3b_3 + 2b_2 + b_1 - c_1, \\ 0 = 6b_3 + 2b_2 - 2c_2, \\ \vartheta_3 = c_4 + c_3 + c_2 + c_1 + c_0, \\ v_3 = 4c_4 + 3c_3 + 2c_2 + c_1, \\ \alpha_3 = 12c_4 + 6c_3 + 2c_2. \end{array} \right.$$

Перепишем систему уравнений в матричном виде

$$\begin{bmatrix} \vartheta_0 \\ v_0 \\ \alpha_0 \\ \vartheta_1 \\ \vartheta_1 \\ 0 \\ 0 \\ \vartheta_2 \\ \vartheta_2 \\ 0 \\ 0 \\ \vartheta_3 \\ v_3 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 3 & 2 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 12 & 6 & 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 3 & 2 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 2 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12 & 6 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix},$$

откуда можем выразить вектор неизвестных коэффициентов

$$\begin{bmatrix} a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 3 & 2 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 12 & 6 & 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 3 & 2 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 2 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12 & 6 & 2 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \vartheta_0 \\ v_0 \\ \alpha_0 \\ \vartheta_1 \\ \vartheta_1 \\ 0 \\ 0 \\ \vartheta_2 \\ \vartheta_2 \\ 0 \\ 0 \\ \vartheta_3 \\ v_3 \\ \alpha_3 \end{bmatrix}.$$

Таким образом, получены полиномы для каждого звена, рассчитаны коэффициенты, с помощью которых может быть осуществлена интерполяция обобщенных координат, скоростей и ускорений при перемещении робота через четыре заданные конфигурации.

## Вопросы для самоконтроля

1. В чем отличие задачи планирования пути от задачи планирования траектории?
2. Назовите достоинства и недостатки метода потенциального поля для планирования маршрута.
3. Сформулируйте особенность методов планирования маршрута, основанных на биоинспирированном подходе. Приведите примеры.
4. Какие ограничения необходимо накладывать на траекторию при решении задачи планирования? Зачем это нужно?
5. Приведите алгоритм решения задачи планирования траектории при заданных промежуточных конфигурациях манипуляционного робота.

## 1.4 Управление движением

Перейдем к задаче синтеза управления движением манипуляционного робота, который представляет собой многомерную систему, т.е. содержит  $n$  управляющих и  $n$  регулируемых переменных. Данная система имеет перекрестные связи, каждое следующее звено влияет на все предыдущие. Таким образом, управление многозвенным манипулятором является нетривиальной задачей. Рассмотрим некоторые подходы к решению поставленной задачи.

### 1.4.1 Одномерное управление

При одномерном управлении момент нагрузки рассматривается как внешнее возмущение (см. рисунок 1.10).

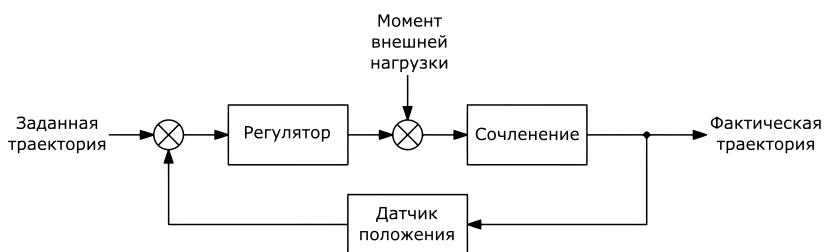


Рисунок 1.10. Общая функциональная схема одномерного управления сочленением

При одномерном управлении нам необходимо решить следующие задачи:

- обеспечить слежение обобщенной координаты за заданным сигналом, определяющим траекторию движения сочленения;
- компенсировать внешнее возмущение, вызванное влиянием гравитационных сил и перекрестных связей звеньев манипулятора.

Независимое управление каждым сочленением манипулятора по сути сводится к управлению электроприводом с зубчатой переда-

чей, динамическая модель которого содержит две составляющие: электрическую и механическую.

Математическую модель вращательного сочленения в виде передаточной функции  $P(s)$  можно представить как [10]

$$P(s) = \frac{1}{s(Js + K)},$$

где  $J$  — обобщенный момент инерции двигателя и зубчатой передачи,  $K = K_f + \frac{K_\varepsilon K_\mu}{R}$ ,  $K_\varepsilon$  — коэффициент противо-ЭДС,  $K_f$  — коэффициент трения,  $K_\mu$  — коэффициент момента силы,  $R$  — сопротивление якоря.

Упрощенная схема моделирования одномерного управления сочленением [10] приведена на рисунке 1.11.

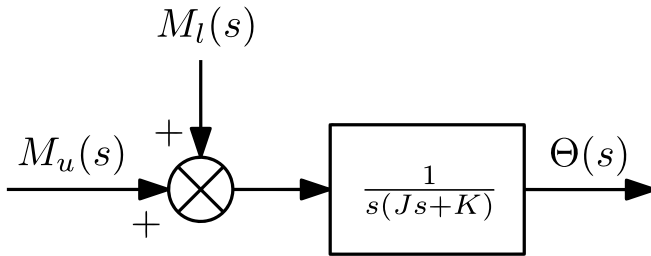


Рисунок 1.11. Упрощенная схема моделирования одномерного управления сочленением

Запишем систему с использованием передаточной функции

$$\Theta(s) = \frac{1}{s(Js + K)}(M_u(s) - M_l(s)) = P(s)(M_u(s) - M_l(s)),$$

где  $M_u(s) = \frac{K_\mu}{R}U(s)$  — управляющий момент силы,  $U(s)$  — управляющее напряжение,  $M_l(s)$  — момент нагрузки.

Синтезируем регулятор для рассматриваемой системы.

**Регулятор** — математический алгоритм (закон управления), обеспечивающий желаемое поведение объекта управления путем достижения заданных свойств устойчивости замкнутой системы.

Введем пропорционально-интегрально-дифференциальный (ПИД) регулятор, заданный передаточной функцией вида

$$R(s) = k_p + k_i \frac{1}{s} + k_d s.$$

Осуществляя структурные преобразования, выразим выходную переменную как

$$\Theta(s) = \frac{R(s)P(s)}{1 + R(s)P(s)} \Theta^*(s) + \frac{P(s)}{1 + R(s)P(s)} M_l(s).$$

Рассчитаем передаточную функцию замкнутой системы

$$W(s) = \frac{R(s)P(s)}{1 + R(s)P(s)} = \frac{\frac{k_d s^2 + k_p s + k_i}{J s^3 + K s^2}}{1 + \frac{k_d s^2 + k_p s + k_i}{J s^3 + K s^2}} = \frac{k_d s^2 + k_p s + k_i}{J s^3 + (K + k_d) s^2 + k_p s + k_i}.$$

Схема моделирования замкнутой системы с ПИД-регулятором приведена на рисунке 1.12. Далее при известных параметрах объекта  $J$  и  $K$  на основании корней характеристического полинома передаточной функции  $J s^3 + (K + k_d) s^2 + k_p s + k_i$  с помощью метода модального управления возможно рассчитать коэффициенты ПИД-регулятора  $k_p$ ,  $k_i$  и  $k_d$  для обеспечения устойчивости замкнутой системы и достижения заданных показателей качества переходных процессов.

## 1.4.2 Многомерное управление

Перейдем к многомерному управлению манипуляционного робота. Для этого необходимо использовать динамическую модель многозвенного робота, представленную в разделе 1.2.

Рассмотрим многомерную динамическую модель манипулятора



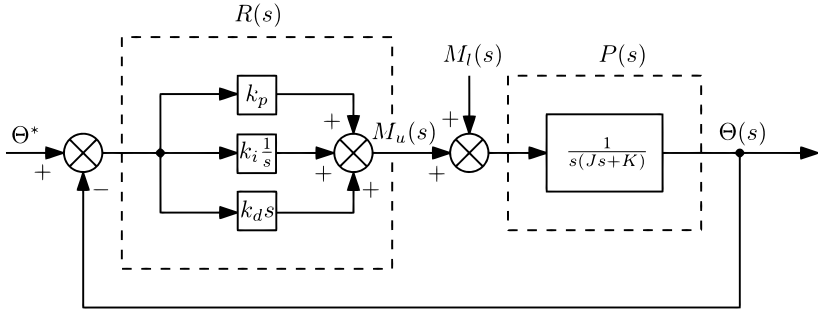


Рисунок 1.12. Схема моделирования замкнутой системы с ПИД-регулятором

вида

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F\dot{q} + G(q) = u,$$

где  $M(q) \in \mathbb{R}^{n \times n}$  — матрица инерции,  $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$  — матрица Кориолиса,  $F \in \mathbb{R}^{n \times n}$  — матрица демпфирования,  $G(q) \in \mathbb{R}^{n \times 1}$  — вектор гравитационных сил.

При проектировании механической части робототехнической системы разработчики минимизируют влияние сил трения в сочленения манипулятора. Поэтому для упрощения последующего синтеза управления и анализа устойчивости будем рассматривать модель без учета сил трения, т.е. при  $F(q) = 0$ . В таком случае динамическая модель манипуляционного робота имеет вид

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = u. \quad (1.32)$$

Одной из самых популярных задач в области теории автоматического управления является слежение за желаемой траекторией  $q^*(t)$  с помощью синтеза нелинейного управления для робототехнической системы.

Идея заключается в выборе такого вектора управляющих воздействий  $u$ , который обеспечит свойство линейности замкнутой системы. Этот метод носит название линеаризация по обратной связи.

Выберем  $u$  в форме

$$u = M(q)u_q + C(q, \dot{q})\dot{q} + G(q), \quad (1.33)$$

где  $u_q$  — новый вектор управляющих воздействий.

Подставим (1.33) в модель (1.32), после чего получим линейную систему

$$\ddot{q} = u_q. \quad (1.34)$$

Выберем новый вектор управляющих воздействий в форме

$$u_q = K_p \tilde{q}(t) + K_d \dot{\tilde{q}}(t) + \ddot{q}^*(t), \quad (1.35)$$

где  $\tilde{q}(t) = q^*(t) - q(t)$ ,  $q(t)$  — действительная траектория.

Подставим управление (1.35) в модель (1.34), после чего получим замкнутую систему

$$\ddot{q} = K_p \tilde{q}(t) + K_d \dot{\tilde{q}}(t) + \ddot{q}^*(t), \quad (1.36)$$

откуда после преобразования получим

$$0 = K_p \tilde{q}(t) + K_d \dot{\tilde{q}}(t) + \ddot{\tilde{q}}(t), \quad (1.37)$$

где матрицы  $K_p$  и  $K_d$  для обеспечения требуемого качества переходного процесса могут быть выбраны в форме

$$K_p = \begin{bmatrix} \omega_1^2 & 0 & \cdots & 0 \\ 0 & \omega_2^2 & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & 0 & \omega_n^2 \end{bmatrix}, \quad (1.38)$$

$$K_d = \begin{bmatrix} 2\omega_1 & 0 & \cdots & 0 \\ 0 & 2\omega_2 & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & 0 & 2\omega_n \end{bmatrix}. \quad (1.39)$$

Таким образом, мы рассмотрели решение задачи слежения за заданной траекторией с помощью линеаризации по обратной связи. Принцип этого подхода проиллюстрирован на рисунке ниже (см.

рисунок 1.13).

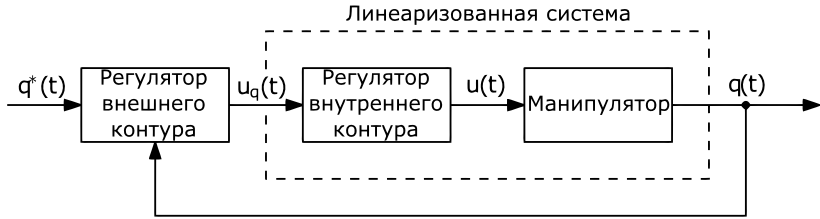


Рисунок 1.13. Принцип линеаризации по обратной связи

Перейдем от управления в конфигурационном пространстве к управлению в рабочем пространстве. Для этого используем аналитическую матрицу Якоби  $J_a(q)$ . Запишем соотношение между скоростями

$$\chi_a(t) = J_a(q)\dot{q}(t), \quad (1.40)$$

где  $\chi_a(t) \in \mathbb{R}^6$  — вектор скоростей вида

$$\chi_a(t) = \begin{bmatrix} \dot{p}_n^0(q) \\ \dot{\eta}_n^0(q) \end{bmatrix}.$$

Возьмем производную от (1.40)

$$\ddot{\chi}_a(t) = \dot{J}_a(q)\dot{q} + J_a(q)\ddot{q}(t),$$

откуда, выразив  $\ddot{q}$  и подставив в (1.34), получим

$$u_q = J_q^{-1}(u_\chi - \dot{J}_a\dot{q}),$$

где  $u_\chi$  — новое управляющее воздействие вида

$$\ddot{\chi}_a(t) = u_\chi.$$

При желаемой траектории, заданной в координатах  $\chi_a^*(t)$ , выберем новый закон управления  $u_\chi$  в форме

$$u_\chi = \ddot{\chi}_a(t) + K_p\tilde{\chi}_a(t) + K_d\dot{\tilde{\chi}}_a(t),$$

где  $\tilde{\chi}_a(t) = \chi_a^*(t) - \chi_a(t)$  — ошибка слежения.

Таким образом, подход с использованием регуляторов внутреннего и внешнего контуров позволяет решать задачу слежения за траекторией, заданной относительно базовой системы координат. Заметим, что для этого необходимым условием является обратимость аналитической матрицы Якоби  $J_a(q)$ .

### 1.4.3 Импедансное управление

Импедансное управление относится к методам косвенного управления силой, поскольку оно обеспечивает контроль над силами контактного взаимодействия с помощью динамического изменения поведения робота без прямых измерений моментов. Управление по импедансу позволяет задать рабочему органу робота некоторую податливость, определяемой виртуальной системой пружина-демпфер, усилия от которой генерируются между схватом и точкой интереса, в которую перемещается рабочий орган. Данный способ управления особо полезен в задачах, где роботу необходимо вести себя подобно человеку, например, при взаимодействии с людьми или выполнении сложных манипуляций.

На рис. 1.14 представлена схема импедансного управления. Закон управления импедансом вычисляет моменты звеньев на основе как желаемого, так и фактического движения (обобщенные координаты и скорости) робота при взаимодействии с окружающей средой. Для работы данного алгоритма не требуется измерение фактического вектора сил контактного взаимодействия, но в большинстве случаев по ряду причин в реальных роботах применяют обратную связь, измеренную с помощью некоторого вида сенсоров, например, датчика силы/момента на сочленении.

Закон управления импедансом записывается путем определения смещения между желаемым и фактическим положением рабочего органа как  $x_d - x_e$

$$\tau = B(q)y + C(q, \dot{q})\dot{q} + F_v\dot{q} + F_s \text{sign}(\dot{q}) + g(q),$$

где  $\tau$  — это вектор управляющих моментов,  $B(q)$  — матрица инерции робота,  $C(q, \dot{q})\dot{q}$  — вектор кориолисовых сил и центробежных моментов,  $F_v\dot{q}$  — вектор моментов вязкого трения,  $g(q)$  — вектор

гравитационных моментов,  $F_s \text{sign}(\dot{q})$  — вектор моментов сухого трения, а  $y$  — вектор разрешенных ускорений для каждого звена

$$y = J_A^{-1}(q)M_d^{-1} \left( M_d \ddot{x}_d + K_d(\dot{x}_d - \dot{x}_e) + K_p(x_d - x_e) - M_d \dot{J}_A(q, \dot{q})\dot{q} \right),$$

где  $J_A(q)$  — это аналитическая матрица Якоби манипулятора, а  $\dot{J}_A(q, \dot{q})$  — его производная по времени.  $M_d$ ,  $K_d$  и  $K_p$  — это желаемая матрица инерции, матрица демпфирования и матрица жесткости желаемого импеданса схвата манипулятора. Напомним, что аналитическая матрица Якоби — это матрица Якоби, связывающая скорость звеньев со скоростью рабочего органа в операционном пространстве, т.е.  $\dot{x}_e = J_A(q)\dot{q}$ . Здесь предполагается, что манипулятор имеет открытую кинематику и аналитическая матрица Якоби  $J_A(q)$  обратима. Таким образом, в данном разделе были рассмотрены некоторые методы управления манипуляционным роботом. Предложена структура для одномерного и многомерного управления манипулятором, решена задача слежения за заданной траекторией и предложена структура для управления по импедансу.

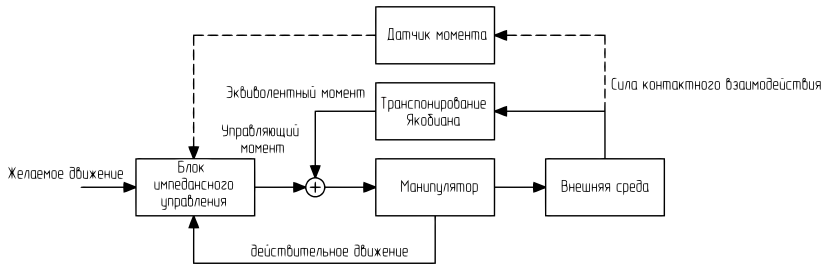


Рисунок 1.14. Общая схема импедансным управлением манипулятивным роботом

## Вопросы для самоконтроля

1. Для чего необходимо исследовать динамическую модель робототехнической системы?
2. Сформулируйте в чем заключается прямая и обратная задача динамики.
3. Запишите уравнение движения манипуляционного робота в матричном виде.
4. В чем заключается физический смысл компонентов уравнения движения?
5. Назовите специальные случаи применения метода Эйлера-Лагранжа.

# Глава 2

## Лабораторные работы

### 2.1 Динамическая модель многозвенного манипулятора

#### Задание

**Цель работы.** Создать и доопределить модель манипулятора и решить задачу обратной динамики.

Для выполнения данной работы поставлены следующие **задачи**.

1. По выбранному варианту кинематики робота (см. таблицу 2.1) загрузить модель манипулятора из программного пакета и вывести параметры Денавита-Хартенберга.
2. Задать массы звеньев, положения их центра масс, тензоры инерции, моменты инерции приводов, коэффициенты вязкого трения приводов, коэффициенты кулоновского трения приводов, передаточные числа редукторов и ограничения по обобщенным координатам. Рекомендуется обратиться к паспортам используемых манипуляторов.
3. Задать и отобразить произвольные начальную и конечную конфигурации робота.
4. Спланировать траекторию между заданными конфигурациями.
5. Решить обратную задачу динамики методом Ньютона-Эйлера для следующих сценариев:

- а) ненулевые скорости и ускорения  $\dot{q} \neq 0, \ddot{q} \neq 0$ ;
  - б) ненулевые скорости и пренебрежимо малые ускорения  $\dot{q} \neq 0, \ddot{q} \approx 0$  — квазистатика;
  - в) нулевые скорости и ускорения  $\dot{q} = 0, \ddot{q} = 0$  — удержание заданного положения.
6. Определить численные значения элементов матриц  $M(q), C(q, \dot{q}), G(q)$  в каждый расчетный момент времени.
  7. Построить графики моментов звеньев манипулятора вдоль траектории для каждого сценария (см. пункт №5).
  8. Составить отчёт в формате .ipynb (см. приложение В).

## Рекомендуемые функции программного пакета:

- `import roboticstoolbox as rtb` — импорт библиотеки `roboticstoolbox`;
- `robot = roboticstoolbox.models.DH.Puma560()` — инициализация объекта (модели в параметрах Денавита-Хартенберга) соответствующего робота; аналогичным образом можно загрузить другие модели: KR5, UR3, UR5, UR10, IRB140, Mico, Jaco;
- `robot.links[i].dyn()` — вывод всех параметров  $i$ -го звена;
- `robot.links[i].m = 1` — задание массы  $i$ -го звена;
- `robot.links[i].r = [1, 2, 3]` — задание положения центра масс  $i$ -го звена;
- `robot.links[i].I = [1, 1, 1, 0, 0, 0]` — задание тензора инерции  $i$ -го звена ( $L_{xx}, L_{yy}, L_{zz}, L_{xy}, L_{yz}, L_{xz}$ );
- `robot.links[i].Jm = 1` — задание момента инерции привода  $i$ -го звена;
- `robot.links[i].B = 1` — задание коэффициента вязкого трения привода  $i$ -го звена;



- `robot.links[i].Tc = 1` — задание коэффициента кулоновского трения привода  $i$ -го звена;
- `robot.links[i].G = 1` — задание передаточного числа  $i$ -го звена;
- `robot.links[i].qlim = [-3.14, 3.14]` — задание ограничений по обобщенным координатам  $i$ -го звена;
- `robot.plot(q)` — отображение конфигурации робота при заданных обобщенных координатах  $q$ ;
- `tr = rtb.jtraj(q_start, q_end, time)` — планирование траектории; `q_start` и `q_end` - массивы начальных и конечных обобщенных координат манипулятора, `time` — одномерный временной массив;
- `tr.q` — обращение к массиву положений каждого звена робота в расчетные моменты времени;
- `tr.qd` — обращение к массиву скоростей каждого звена робота в расчетные моменты времени;
- `tr.qdd` — обращение к массиву ускорений каждого звена робота в расчетные моменты времени;
- `tau = robot.rne(tr.q, tr.qd, tr.qdd)` — решение обратную задачу динамики;
- `M = robot.inertia(tr.q)` — вычисление матрицы сил инерции;
- `C = robot.coriolis(tr.q, tr.qd)` — вычисление матрицы центробежных сил;
- `G = robot.gravload(tr.q)` — вычисление матрицы потенциальных сил.

## 2.2 Планирование траектории многозвенного манипулятора

### Задание

**Цель работы.** Решить задачу планирования траектории для многозвенного манипуляционного робота.

Для выполнения данной работы поставлены следующие **задачи**.

1. По выбранному варианту кинематики робота (см. таблицу 2.1) загрузить модель манипулятора из программного пакета.
2. Заполнить все параметры модели робота согласно лабораторной работе №1.
3. Задать произвольную начальную конфигурацию робота.
4. Решить прямую задачу кинематики при заданных обобщенных координатах.
5. Построить рабочую зону манипулятора при заданных ограничениях в сочленениях робота.
6. Выбрать конечную точку из рабочей зоны и решить для нее обратную задачу кинематики.
7. Спланировать траекторию между начальным и конечным положением схвата манипулятора с использованием минимум трех методов планирования.
8. Вывести графики изменения положения, скорости и ускорения звеньев робота при движении вдоль траектории.
9. Составить отчет в формате .irub (см. приложение В).

№ варианта	Модель манипулятора
1	KR 5
2	UR3
3	IRB140
4	UR5
5	UR10
6	Kinova MICO
7	Kinova JACO

Таблица 2.1. Варианты заданий

### Рекомендуемые функции программного пакета:

- `import spatialmath as sm` — импорт библиотеки `spatialmath`;
- `tr = rtb.mtraj(rtb.trapezoidal, q_start, q_end, time)` — планирование траектории; `rtb.trapezoidal` (или `rtb.quintic`) — метод планирования траектории;
- `T = robot.fkine(q)` — решение прямой задачи кинематики; `q` — массив обобщённых координат, `T` —  $SE(3)$  матрица;
- `T = sm.base.transl([x,y,z])` — формирование  $SE(3)$  матрицы из координат `x`, `y`, `z`;
- `q = robot.ikine_GN(T).q` — решение обратной задачи кинематики методом Гаусса-Ньютона (в программном пакете реализованы и другие методы).

## 2.3 Управление манипуляционным роботом

### Задание

Для выполнения данной работы поставлены следующие **задачи**.

1. По выбранному варианту кинематики робота (см. таблицу 2.1) загрузить модель манипулятора из программного пакета.
2. Заполнить все параметры модели робота согласно лабораторной работе №1.
3. Задать желаемую траекторию движения манипулятора согласно лабораторной работе №2.
4. Создать модель системы управления роботом, реализующую регулирование на основе обратной модели динамики. В работе рекомендуется синтезировать ПД-регулятор (допустимо использование других регуляторов).
5. Настроить коэффициенты регулятора, минимизирующие ошибку обобщенных координат звеньев.
6. Вывести графики изменения положения, скорости и момента для каждого из звеньев робота при движении вдоль траектории.
7. Добавить полезную нагрузку и повторить пункты № 5-6.
8. Составить отчет в формате .irupb (см. приложение В).

## Рекомендуемые функции программного пакета:

- `def regfunc(robot, t, q_act, qd_act) ... return tau` — пользовательская функция регулятора; `t` — текущий момент времени; `q_act` — вектор значений обобщенных координат манипулятора в момент времени `t`; `qd_act` — вектор значений обобщенных скоростей манипулятора в момент времени `t`; `tau` — вектор моментов в момент времени `t`;
- `tg = robot.fdyn(t_stop, q_start, regfunc, dt=t_shag, progress=True, solver="RK45", solver_args={"atol":1e-3,"rtol":1e-2})` — решение прямой задачи динамики; `t_stop` — время симуляции; `q_start` — начальное положение манипулятора; `dt` — шаг дискретизации; `progress` — отображение прогресса вычислений; `solver` — выбор решателя (RK45, RK23, LSODA и т.д.), его можно подобрать эмпирически; `solver_args` — параметры решателя; `"atol"` и `"rtol"` — абсолютный и относительный допуски, их значения тоже можно подобрать эмпирически;
- `robot.payload(m,p)` — добавление полезной нагрузки `m`;  
`p = array([x,y,z])` — положение в системе координат энд-эффектора.

## 2.4 Комбинированные регуляторы для траекторного управления и непрямого управления по силе

### Задание

Для выполнения данной работы поставлены следующие **задачи**.

1. По выбранному варианту кинематики робота (см. таблицу 2.1) загрузить модель манипулятора из программного пакета.
2. Заполнить все параметры модели робота согласно лабораторной работе №1.
3. Задать желаемую траекторию движения манипулятора согласно лабораторной работе №2.
4. Создать модель системы управления роботом, реализующую регулирование на основе импедансного управления.
5. Настроить коэффициенты регулятора, минимизирующие ошибку положения схвата.
6. Вывести графики изменения положения, скорости и момента для каждого из звеньев робота при движении вдоль траектории.
7. Добавить полезную нагрузку и повторить пункты №5 и №6.
8. Составить отчёт в формате .irunb (см. приложение В).

## Заключение

Учебно-методическое пособие посвящено управлению многозвенными робототехническими системами. Первая глава посвящена теоретическим аспектам, необходимым для освоения дисциплины "Моделирование и управление движением роботов". В четырех разделах первой главы представлен материал, необходимый для выполнения лабораторных и курсовых проектов.

Первый раздел первой главы посвящен кинематическому анализу робототехнической системы. Для описания используется кинематическая схема, решаются прямая и обратная задачи кинематики, которые помогают найти связи между положениями систем координат базы робота и рабочего органа. Также рассмотрен вопрос нахождения связи между скоростями обобщенных координат и рабочего органа, как линейных, так и угловых составляющих.

Второй раздел первой главы сосредоточен на динамическом анализе манипуляционных роботов, рассмотрены прямая и обратная задача динамики, представлена динамическая модель манипуляционного робота, описаны параметры звеньев робота, необходимые для проведения динамического анализа.

Третий раздел первой главы развивает полученные подходы для решения задачи планирования движения робота. Данная задача осложняется конфигурацией самого робота и связями между его звеньями, а также конфигурацией рабочего пространства, которое может иметь дополнительные запрещенные зоны, вызванные особенностями технологического процесса. Представлено несколько подходов к решению данной задачи.

Четвертый раздел первой главы уделяет внимание синтезу управления многозвенной робототехнической системой с целью следова-

ния за траекторией, которая была рассчитана с помощью решений, представленных в предыдущих трех частях.

Вторая глава содержит задания для выполнения четырех лабораторных работ и полезные функции программного пакета, которые можно использовать для выполнения поставленных задач.

Приложение к учебно-методическому пособию состоит из двух частей. В Приложении А представлено задание, критерии оценивания и предполагаемые темы для выполнения курсовых проектов. В приложении Б указаны все команды программного пакета, необходимые для выполнения лабораторных работ в рамках дисциплины в алфавитном порядке.



# Приложение А. Задание для курсового проекта

**Целью** курсового проекта является освоение знаний о типовых кинематических схемах робототехнических систем, принципах моделирования робототехнической системы с применением CRT, а также синтеза системы управления. Для выполнения поставленной цели необходимо выполнить следующие **задачи**:

- провести кинематический анализ робототехнической системы;
- построить динамическую модель робота;
- разработать планировщик траекторий, согласно возможному сценарию функционирования системы;
- синтезировать регулятор для управления движением робота.
- провести Численное моделирование;
- составить пояснительную записку по ГОСТ 7.32-2017.

Для выполнения работы требуется выбрать один из предложенных вариантов робототехнической системы или предложить свой согласно тематике выпускной квалификационной работы. Все неуказанные геометрические параметры, необходимые для работы, требуется задать самостоятельно. Значения геометрии, массы, моментов инерции необходимо указать или рассчитать самостоятельно. При планировании движения необходимо решить задачу планирования пути и планирования траектории. Желаемая траектория должна иметь сложную форму (непрямая линия, окружность) с несколькими переломами (первая производная от траектории знакопеременна). В работе необходимо описать все ограничения и до-

пушения, которые учитывались при синтезе системы управления. В таблице 2.2 представлены возможные задачи для предложенных робототехнических систем. Используются следующие обозначения ✓ — обязательная задача, O — задача по выбору (необходимо выбрать одну из задач). Объем отчета должен быть не менее 10 и не более 20 страниц.

## Оценивание курсового проекта

### Отлично

1. Исследование выполнено самостоятельно, имеет научно-практический характер, содержит элементы новизны.
2. Обучающийся показал знание теоретического материала по рассматриваемой проблеме, умение анализировать, аргументировать свою точку зрения, делать обобщение и выводы.
3. Материал излагается грамотно, логично, последовательно.
4. Оформление отвечает требованиям написания курсовой работы/ проекта.
5. Во время защиты обучающийся показал умение кратко, доступно (ясно) представить результаты исследования, адекватно ответить на поставленные вопросы.

### Хорошо

1. Исследование выполнено самостоятельно, имеет научно-практический характер, содержит элементы новизны.
2. Обучающийся показал знание теоретического материала по рассматриваемой проблеме, однако умение анализировать, аргументировать свою точку зрения, делать обобщения и выводы вызывают у него затруднения.
3. Материал не всегда излагается логично, последовательно.
4. Имеются недочеты в оформлении курсовой работы/ проекта.

5. Во время защиты обучающийся показал умение кратко, доступно (ясно) представить результаты исследования, однако затруднялся отвечать на поставленные вопросы.

### **Удовлетворительно**

1. Исследование не содержит элементы новизны.
2. Обучающийся не в полной мере владеет теоретическим материалом по рассматриваемой проблеме, умение анализировать, аргументировать свою точку зрения, делать обобщение и выводы вызывают у него затруднения.
3. Материал не всегда излагается логично, последовательно.
4. Имеются недочеты в оформлении курсовой работы/ проекта.
5. Во время защиты обучающийся затрудняется в представлении результатов исследования и ответах на поставленные вопросы.

### **Неудовлетворительно**

1. Выполнено менее 50% требований к курсовому проекту.

## Варианты проектов

Робототехническая система/ Тип задачи управления	Кинематический анализ	Динамический анализ	Планирование движения	Слежение за траекторией	Стабилизация
Платформа A1	✓	✓	✓	✓	
Платформа A2	✓	✓	✓	✓	
Платформа A3	✓	✓	✓	✓	
Квадрокоптер A4	✓	✓	✓	○	○
Надводное судно A5	✓	✓	✓	○	○
Гексапод A6	✓	✓	✓	○	○
Двуногий робот A7	✓	✓	✓	○	○
Четырехногий робот A8	✓	✓	✓	○	○
Манипулятор A9	✓	✓	✓	○	

Таблица 2.2. Варианты курсовых проектов

### **A1. Четырёхколёсная платформа с двумя передними приводными поворотными мотор-колёсами**

На рисунке 2.1 представлена кинематическая схема роботизированной платформы с двумя передними приводными поворотными

мотор-колесами.

Здесь  $\omega_{BL}, \omega_{BR}, \omega_{FL}, \omega_{FR}$  — угловые скорости каждого колеса,  $\phi_F$  — угол рыскания колеса, ICR — мгновенный центр вращения.

Необходимо обеспечить перемещение робота по желаемой траектории с ограничением на то, что передние колёса имеют ограничение на поворот в 30 градусов от первоначального положения.

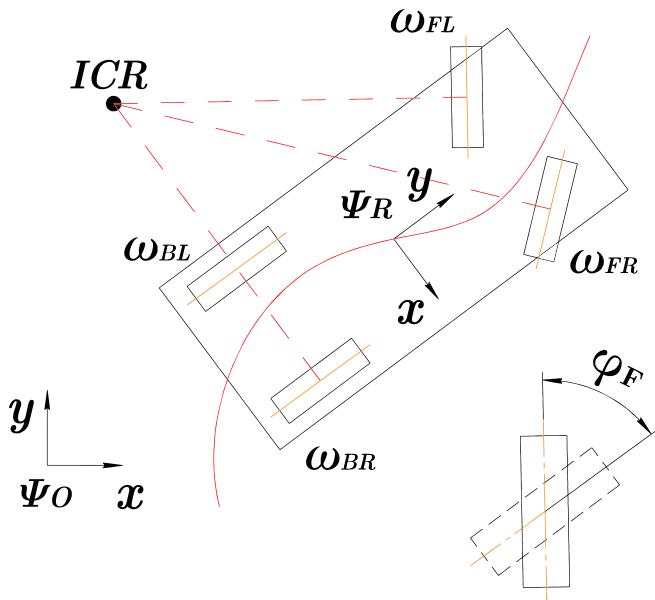


Рисунок 2.1. Кинематическая схема мобильного робота

## А2. Четырёхколёсная платформа с роликонесущими колёсами

На рисунке 2.2 представлена кинематическая схема роботизированной платформы на базе роликонесущих колёс. Особая конструкция колёса позволяет роботам на базе таких движителей совершать движения сложной механической природы.

Здесь  $v_x, v_y$  — линейные скорости,  $\omega_z$  — угловая скорость вращения,  $\alpha_i$  — угол направления вектора скорости платформы  $i$ -го колёса,  $\theta$  — угол поворота платформы,  $l_{iy}, l_{ix}$  — смещения оси колёса от центра платформы,  $\gamma_i$  — угол между роликом и плоскостью колёса. Необходимо обеспечить перемещение робота по желаемой траектории, управляя положением и ориентацией платформы.

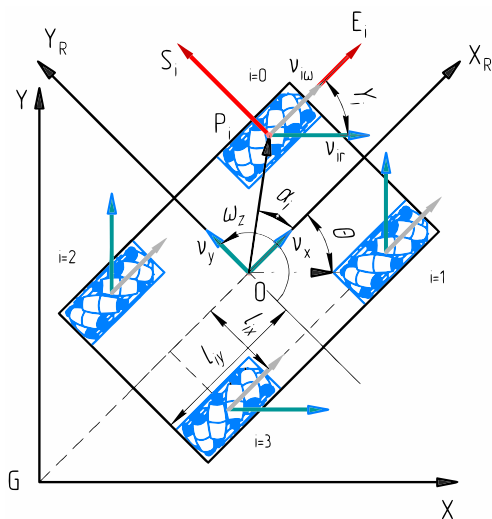


Рисунок 2.2. Кинематическая схема Месашип платформы

### А3. Мобильный робот с дифференциальным приводом

На рисунке 2.3 представлена кинематическая схема роботизированной платформы на базе дифференциального привода с двумя приводными колесами и одним пассивным.

Здесь  $v_y$  — линейная скорость,  $1_f, 2_f, 3_c$  — приводные и пассивное колеса,  $\eta_{1,2}$  — частота вращения,  $L$  — расстояние от центра платформы до оси активных колес,  $l$  — проекция расстояния от центра платформы до оси пассивного колеса,  $\alpha$  — ведомый угол поворота пассивного колеса

Необходимо обеспечить перемещение робота по желаемой траектории с ограничением на поворот пассивного колеса в 40 градусов от первоначального положения.

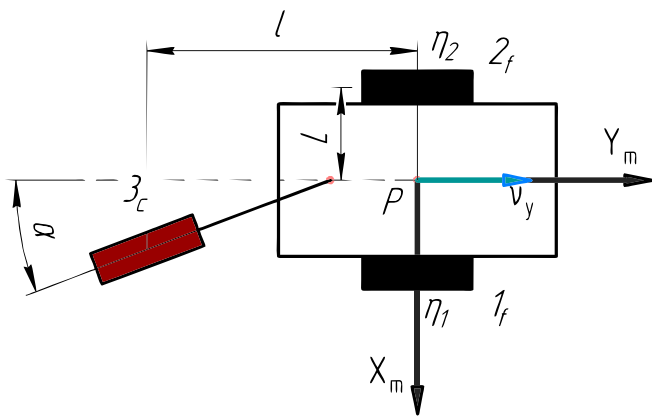


Рисунок 2.3. Кинематическая схема дифференциального робота

## А4. Воздушный робот типа "квадрокоптер"

На рисунке 2.4 представлена кинематическая схема воздушного робота типа «квадрокоптер».

Здесь  $\omega_{1...4}$  — угловые скорости роторов,  $a, b$  — положение центра масс камеры массой  $m$  относительно остальной части массой  $M$ .

Необходимо обеспечить перемещение робота в пространстве по желаемой траектории.

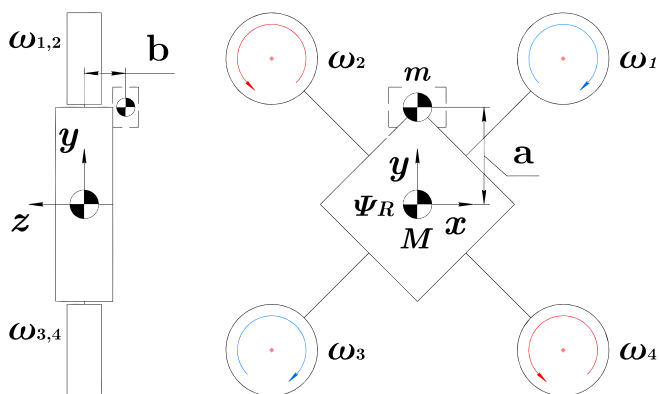


Рисунок 2.4. Кинематическая схема "квадрокоптера"



## А5. Роботизированное надводное судно

На рисунке 2.5 представлена кинематическая схема роботизированного надводного судна. Конфигурация робота включает два независимо управляемых винта, два руля, при этом можно упростить задачу и считать, что оба руля связаны с одним сервоприводом и подруливающее устройство.

Здесь  $\alpha_{l,r}$  — угол поворота левого/правого руля или кормового азимутального трастера,  $l_e$  — расстояние от центра масс до двигателей или азимутальных трастеров, расположенных в корме судна,  $l_b$  — расстояние от центра масс до носового органа управления в продольном направлении,  $P_b$  — сила (упор), развиваемая носовым трастером,  $P_{l,r}$  — сила (упор), развиваемая левым/правым двигателем или кормовым, азимутальным трастером,  $v_x, v_y$  — линейные скорости,  $\omega_z$  — угловая скорость рыскания,  $L_{l,r}$  — осевое смещение двигателя относительно оси симметрии судна.

Необходимо обеспечить перемещение робота в пространстве по желаемой траектории. Можно считать, что управление реализуется в режиме динамического позиционирования.

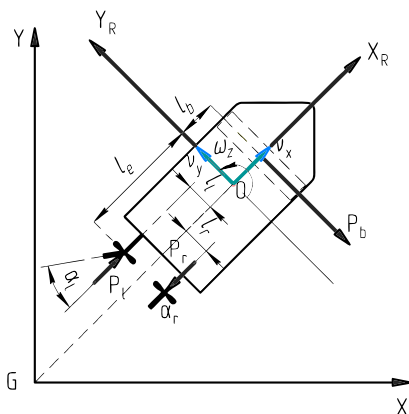


Рисунок 2.5. Кинематическая схема роботизированного надводного судна

## А6. Шестиподвижная платформа

На рисунке 2.6 представлена кинематическая схема шестиподвижной платформы. Шестиподвижная платформа позволяет определить положение и ориентацию энд-эффектора.

Здесь  $l_{1,2}$  — длины звеньев. Необходимо обеспечить движение по траектории в пространстве.

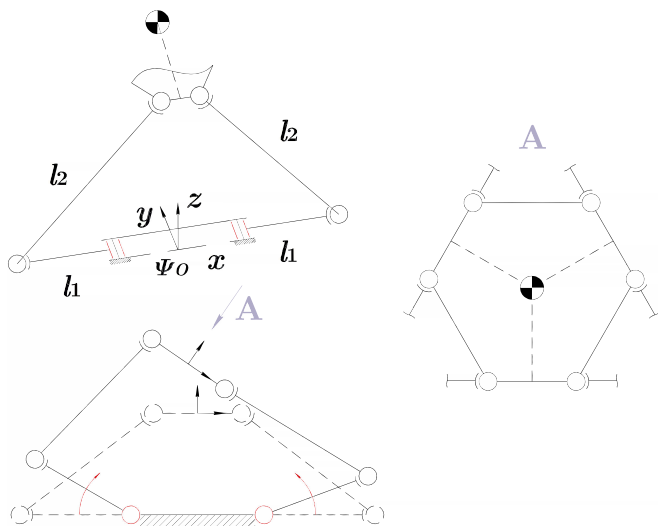


Рисунок 2.6. Кинематическая схема платформы

## А7. Двухногий плоскорычажный робот

На рисунке 2.7 представлен двухногий робот, каждая нога которого составлена из двух звеньев. Коленный сустав ноги подпружинен и позволяет компенсировать массу робота.

Здесь  $k$  — крутильная жесткость пружины,  $m_h, m_s, M$  — массы бедра, голени и туловища робота соответственно,  $L_h, L_s, B$  — длины бедра, голени и туловища соответственно,  $L_0, H_0$  — габариты препятствия. Требуется решить следующие две задачи.

1. Определить жёсткость пружин  $k$ , компенсирующих вес робота.
2. Составить оптимальное управление приводами робота таким образом, чтобы он мог перепрыгнуть препятствие, не касаясь его, потратив при этом минимальное количество энергии на двигателях.

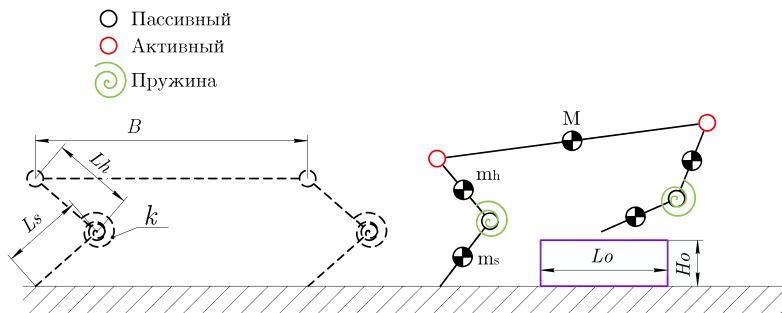


Рисунок 2.7. Кинематическая схема робота

## А8. Четырехногий робот

На рисунке 2.8 представлен четвероногий робот, каждая нога которого составлена из двух звеньев.

Здесь  $m_h, m_s, M$  — массы бедра, голени и туловища робота соответственно,  $L_h, L_s, L_a, L_b$  — длины бедра, голени и габариты туловища соответственно. Требуется синтезировать управление приводами робота, выполняющего перемещение на плоскости по заданной траектории.

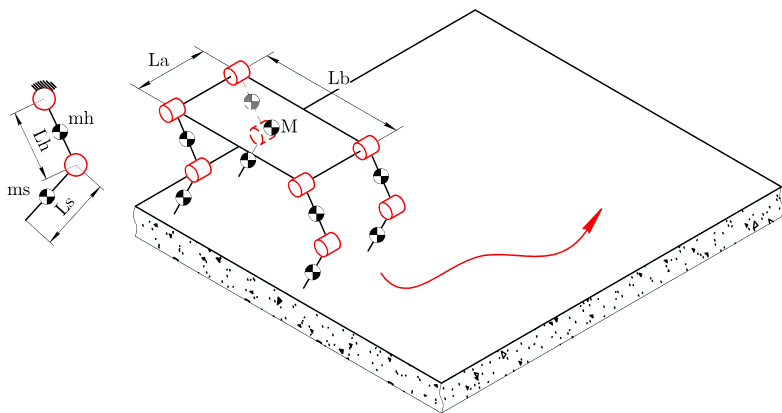


Рисунок 2.8. Кинематическая схема робота

## А9. Манипуляционный робот с избыточной кинематикой

На рисунке 2.9 представлен шарнирный манипуляционный робот с избыточной кинематикой, состоящий из семи звеньев.

Здесь  $m_{1...7}$ ,  $l_{1...7}$  — массы и длины звеньев соответственно. Необходимо провести кинематический анализ, рассчитать, подобрать или присвоить все ключевые параметры звеньев и приводов в сочленениях, спланировать траекторию и разработать алгоритм управления для перемещения вдоль заданной траектории.

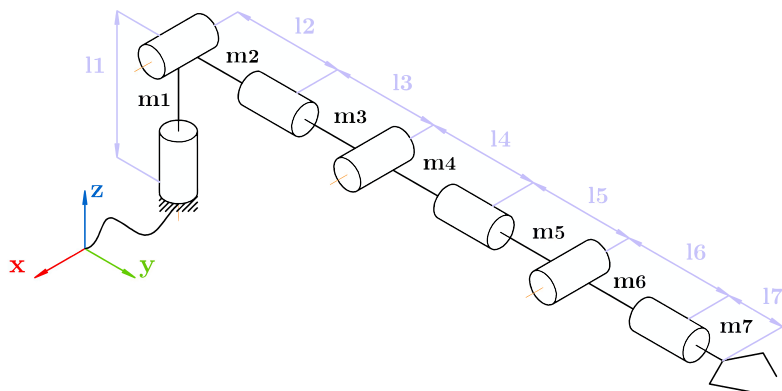


Рисунок 2.9. Кинематическая схема робота

## Приложение Б. Команды для Corke robotics toolbox в Python

- `C = robot.coriolis(tr.q, tr.qd)` — вычисление матрицы центробежных сил;
- `def regfunc(robot, t, q_act, qd_act) ... return tau` — пользовательская функция регулятора; `t` — текущий момент времени; `q_act` — вектор значений обобщенных координат манипулятора в момент времени `t`; `qd_act` — вектор значений обобщенных скоростей манипулятора в момент времени `t`; `tau` — вектор моментов в момент времени `t`;
- `G = robot.gravload(tr.q)` — вычисление матрицы потенциальных сил;
- `import roboticstoolbox as rtb` — импорт библиотеки `roboticstoolbox`;
- `import spatialmath as sm` — импорт библиотеки `spatialmath`;
- `M = robot.inertia(tr.q)` — вычисление матрицы сил инерции;
- `q = robot.ikine_GN(T).q` — решение обратной задачи кинематики методом Гаусса-Ньютона (в программном пакете реализованы и другие методы);
- `robot = roboticstoolbox.models.DH.Puma560()` — инициализация объекта (модели в параметрах Денавита-Хартенберга) соответствующего робота; аналогичным

образом можно загрузить другие модели: KR5, UR3, UR5, UR10, IRB140, Mico, Jaco;

- `robot.links[i].B = 1` — задание коэффициента вязкого трения привода  $i$ -го звена;
- `robot.links[i].dyn()` — вывод всех параметров  $i$ -го звена;
- `robot.links[i].G = 1` — задание передаточного числа  $i$ -го звена;
- `robot.links[i].I = [1, 1, 1, 0, 0, 0]` — задание тензора инерции  $i$ -го звена ( $L_{xx}$ ,  $L_{yy}$ ,  $L_{zz}$ ,  $L_{xy}$ ,  $L_{yz}$ ,  $L_{xz}$ );
- `robot.links[i].Jm = 1` — задание момента инерции привода  $i$ -го звена;
- `robot.links[i].m = 1` — задание массы  $i$ -го звена;
- `robot.links[i].qlim = [-3.14, 3.14]` — задание ограничения по обобщенным координатам  $i$ -го звена;
- `robot.links[i].r = [1, 2, 3]` — задание положения центра масс  $i$ -го звена;
- `robot.links[i].Tc = 1` — задание коэффициента кулоновского трения привода  $i$ -го звена;
- `robot.payload(m,p)` — добавление полезной нагрузки  $m$ ;  
`p = array([x,y,z])` — положение в системе координат энд-эффектора;
- `robot.plot(q)` — отображение конфигурации робота при заданных обобщённых координатах  $q$ ;
- `T = robot.fkine(q)` — решение прямой задачи кинематики;  
 $q$  — массив обобщённых координат,  $T$  — SE(3) матрица;
- `T = sm.base.transl([x,y,z])` — формирование SE(3) матрицы из координат  $x$ ,  $y$ ,  $z$ ;
- `tau = robot.rne(tr.q, tr.qd, tr.qdd)` — решение обратной задачи динамики;

- `tg = robot.fdyn(t_stop, q_start, regfunc, dt=t_shag, progress=True, solver="RK45", solver_args={"atol":1e-3,"rtol":1e-2})` — решение прямой задачи динамики; `t_stop` — время симуляции; `q_start` — начальное положение манипулятора; `dt` — шаг дискретизации; `progress` — отображение прогресса вычислений; `solver` — выбор решателя (RK45, RK23, LSODA и т.д.), его можно подобрать эмпирически; `solver_args` — параметры решателя; `"atol"` и `"rtol"` — абсолютный и относительный допуски, их значения тоже можно подобрать эмпирически;
- `tr = rtb.jtraj(q_start, q_end, time)` — планирование траектории; `q_start` и `q_end` — массивы начальных и конечных обобщённых координат манипулятора, `time` — одномерный временной массив;
- `tr = rtb.mtraj(rtb.trapezoidal, q_start, q_end, time)` — планирование траектории; `rtb.trapezoidal` (или `rtb.quintic`) — метод планирования траектории;
- `tr.q` — обращение к массиву положений каждого звена робота в расчетные моменты времени;
- `tr.qd` — обращение к массиву скоростей каждого звена робота в расчетные моменты времени;
- `tr.qdd` — обращение к массиву ускорений каждого звена робота в расчетные моменты времени.



## **Приложение В. Правила оформления отчетов**

### **Правила оформления отчетов по лабораторным работам**

Отчеты по лабораторным работам оформляются в формате jupyter notebook (файл с расширением .ipynb). По всем этапам лабораторной работы необходимо представить развернутые комментарии.

Структура отчета:

- название лабораторной работы;
- ФИО и номер группы студента;
- цель и задачи лабораторной работы;
- решение каждой задачи лабораторной работы с подробными комментариями;
- графики, полученные в ходе работы;
- выводы о проделанной работе.

### **Правила оформления отчетов по курсовым работам**

Отчет по курсовой работе оформляется в соответствии с ГОСТ 7.32-2017. Объем отчета должен быть не менее 10 и не более 20 страниц. Курсовая работа должна быть представлена не позднее даты сдачи экзамена. Защита курсовой работы предполагается в устной форме.

Структура отчета:

- титульный лист, включающий в себя название работы, ФИО и номер студента(-ов), ФИО преподавателя, дату выполнения работы;
- введение, включающее в себя цель и задачи курсовой работы;
- кинематический анализ рассматриваемой робототехнической системы;
- построение динамической модели рассматриваемой робототехнической системы;
- разработку планировщика движений;
- синтез регулятора для управления движением робота;
- численное моделирование замкнутой системы;
- графики, полученные в ходе работы;
- заключение;
- список использованной литературы;
- приложение с полным листингом программы.

# Литература

- [1] АГ Кравцов and KB Марусич. *Промышленные роботы: учебное пособие*. ОГУ, 2019.
- [2] Peter Corke. *Robotics, Vision and Control*. Springer Cham, 2023.
- [3] Максим Архипов, Михаил Вартанов, and Роман Мищенко. *Промышленные роботы: управление манипуляционными роботами 2-е изд., испр. и доп. Учебное пособие для СПО*. Litres, 2020.
- [4] Евгений Иванович Юревич. *Основы робототехники, 4 изд.* БХВ-Петербург, 2018.
- [5] J. Zhang, S. Wang, H. Wang, J. Lai, Z. Bing, Y. Jiang, Y. Zheng, and Z. Zhang. An adaptive approach to whole-body balance control of wheel-bipedal robot ollie. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12835–12842, 2022.
- [6] ИИ Артоболевский. *Теория машин и механизмов*. М. Наука, 1988.
- [7] Алексей Лесков, Ксения Бажинова, and Елена Селиверстова. *Кинематика и динамика исполнительных механизмов манипуляционных роботов*. МГТУ им. Н.Э. Баумана, 2017.
- [8] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot modeling and control*. John Wiley & Sons, 2020.
- [9] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*. Springer, 2008.

- [10] О. И. Борисов, В. С. Громов, and А. А. Пыркин. *Методы управления робототехническими приложениями*. Университет ИТМО, 2016.
- [11] ВП Умнов and ЮЕ Мишулин. *Моделирование динамики механизмов и управляемого движения исполнительных устройств манипуляционных роботов: учебное пособие*. Владимир: Изд-во ВлГУ, 2019.
- [12] Елена Евгеньевна Ступина, Дмитрий Юрьевич Чупин, Роман Владимирович Каменев, and Андрей Анатольевич Ступин. *Основы робототехники: учебное пособие для студентов вузов*. Новосибирск: Сибпринт, 2019.
- [13] Сергей Алексеевич Колюбин. *Динамика робототехнических систем*. Университет ИТМО, 2017.
- [14] Han-Pang Huang and Shu-Yun Chung. Dynamic visibility graph for path planning. *Proceedings of 2004 IEEE International Conference on Intelligent Robots and Systems*, pages 2813–2818, 2004.
- [15] J. O. Wallgrün. Voronoi graph matching for robot localization and mapping. *Transactions on Computational Science*, 9:76–108, 2010.
- [16] M. K. Habib and H. Asama. Efficient method to generate collision free paths for an autonomous mobile robot based on new free space structuring approach. In *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, volume 2, pages 563–567, 1991.
- [17] A. M. Ladd and L. E. Kavraki. Measure theoretic analysis of probabilistic path planning. *IEEE Transactions on Robotics and Automation*, 20:229–242, 2004.
- [18] D. Floreano, P. Husbands, and S. Nolfi. *Springer Handbook of Robotics*. Springer Verlag, 2008.
- [19] S. Koenig, M. Likhachev, and D. Furcy. Lifelong planning a\*. *Artificial Intelligence*, 155:93–146, 2004.

- [20] L. De Filippis, G. Guglieri, and F. Quagliotti. A minimum risk approach for path planning of uavs. *Journal of Intelligent and Robotic Systems*, 61:203–219, 2011.
- [21] M. Jefferies and Wai-Kiang Yeap. *Robotics and Cognitive Approaches to Spatial Mapping*, volume 38. Springer Berlin, Heidelberg, 2008.
- [22] D. Alvarez, J. Gómez, S. Garrido, and L. Moreno. 3d robot formations path planning with fast marching square. *Journal of Intelligent and Robotic Systems*, 80:507–523, 2015.
- [23] L. De Filippis, G. Guglieri, and F. Quagliotti. Path planning strategies for uavs in 3d environments. *Journal of Intelligent and Robotic Systems*, 65:247–264, 2012.
- [24] M. W. Spong. Robot modeling and control. *Industrial Robot: An International Journal*, 33:403–403, 2006.
- [25] Владимир Олегович Антонов, Михаил Михайлович Гурчинский, Вячеслав Иванович Петренко, and Фариза Биляловна Тебуева. Метод планирования оптимальной траектории движения трехзвенного манипулятора в объемном пространстве с препятствием. *Вестник Дагестанского государственного технического университета. Технические науки*, 45(1):98–112, 2018.
- [26] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *Systems, Man and Cybernetics, IEEE Transactions on*, 19:1179–1187, 1989.
- [27] J. Borenstein and Y. Koren. The vector field histogram - fast obstacle avoidance for mobile robots. *Robotics and Automation, IEEE Transactions on*, 7:278–288, 1991.
- [28] Y. Kuwata. *Real-time Trajectory Design for Unmanned Aerial Vehicles using Receding Horizon Control*. MIT, 2003.
- [29] G. Habibi, E. Masehian, and M. Beheshti. Binary integer programming model of point robot path planning. In *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*, pages 2841–2845, 2007.

- [30] R. Glasius, A. Komoda, and S. Gielen. Neural network dynamics for path planning and obstacle avoidance. *Neural Networks*, 8:125–133, 1995.
- [31] J. Moreno and M. Castro. Heuristic algorithm for robot path planning based on a growing elastic net. In *Progress in Artificial Intelligence*, volume 3808, pages 447–454. Springer Berlin Heidelberg, 2005.
- [32] К. Фу, Р. Гонсалес, and К. Ли. *Робототехника*. Мир, 1989.

Захаров Д.Н., Куровский Д.М., Ракшин Е.А.,  
Борисов О.И., Громов В.С., Колубин С.А.

**МОДЕЛИРОВАНИЕ И УПРАВЛЕНИЕ  
ДВИЖЕНИЕМ РОБОТОВ**

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе

**Редакционно-издательский отдел**  
**Университета ИТМО**  
197101, Санкт-Петербург, Кронверкский пр., 49, литер А